



УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
Штип

М-Р БИСЕРКА ПЕТРОВСКА

**КЛАСИФИКАЦИЈА НА СЦЕНИ СНИМЕНИ ОД ВОЗДУХ СО МЕТОДИТЕ
НА ПРЕНОС НА УЧЕЊЕ ОД КОНВОЛУЦИСКИ НЕВРОНСКИ МРЕЖИ**

- ДОКТОРСКА ДИСЕРТАЦИЈА-

Ментор:
ред. проф. д-рТатјана Атанасова-Пачемска

Кандидат:
Бисерка Петровска
број на индекс 31013

Јуни 2021 година
Штип

Интерен ментор Татјана Атанасова-Пачемска
Редовен професор, Факултет за информатика,
Универзитет „Гоце Делчев“ - Штип

Екстерен ментор Томе Димовски
Вонреден професор, Факултет за информатички и
комуникациски технологии,
Универзитет „Св. Климент Охридски“ - Битола

Комисија за оценка и одбрана:

Претседател Зоран Здравев
Редовен професор, Факултет за информатика,
Универзитет „Гоце Делчев“ - Штип

Член Цвета Мартиновска-Банде
Редовен професор, Факултет за информатика,
Универзитет „Гоце Делчев“ - Штип

Член Наташа Стојковиќ
Вонреден професор, Факултет за информатика,
Универзитет „Гоце Делчев“ - Штип

Научно поле Компјутерска техника и информатика

Научна област Вештачка интелигенција и системи

датум на одбрана -----

датум на промоција -----

Рецензирани и објавени трудови произлезени од истражувањето и презентации од научни собири:

1. Petrovska B. & Atanasova-Pacemska T. (2021). Computer Vision in Aerial Scene Recognition. South-East European Journal of Sustainable Development, Vol.5 (1/2021)
2. Petrovska B., Atanasova-Pacemska T., Stojkovik N., Stojanova A. & Kocaleva M. (2021). Machine Learning with Remote Sensing Image Datasets. Informatica, in editing
3. Petrovska B. & Atanasova-Pacemska T. (2020). Computer Vision in Aerial Scene Recognition. 4th International Conference Towards Sustainable Development TSD, Skopje, North Macedonia, Book of Abstracts.
4. Petrovska B., Zdravevski E., Lameski P., Corizzo R., Štajduhar I. & Lerga J. (2020). Deep Learning for Feature Extraction in Remote Sensing: A Case-Study of Aerial Scene Classification. Sensors, 20, 3906, doi:10.3390/s20143906.
5. Petrovska B., Atanasova-Pacemska T., Corizzo R., Mignone P., Lameski P. & Zdravevski E. (2020). Aerial Scene Classification through Fine-Tuning with Adaptive Learning Rates and Label Smoothing. Appl. Sci., 10, 5792, doi:10.3390/app10175792.
6. Petrovska B., Stojanovic I. & Atanasova-Pacemska T. (2018). Classification of Small Sets of Images with Pre-trained Neural Networks. International Journal of Engineering and Manufacturing (IJEM), Vol.8, No.4, pp.40-55, doi: 10.5815/ijem.2018.04.05.
7. Petrovska B., Stojanovic I. & Atanasova-Pacemska T. (2018). Classification of Small Datasets of Images with Transfer Learning in Convolutional Neural Networks. Balkan Journal of Applied Mathematics and Informatics (BJAMI), Vol.1, No.1, pp.17-24, ISSN: 2545-479X

КЛАСИФИКАЦИЈА НА СЦЕНИ СНИМЕНИ ОД ВОЗДУХ СО МЕТОДИТЕ НА ПРЕНОС НА УЧЕЊЕ ОД КОНВОЛУЦИСКИ НЕВРОНСКИ МРЕЖИ

Апстракт:

Класификацијата на сцени која се базира на слики е есенцијална во многу системи и апликации поврзани со набљудувањето на далечина (remote sensing). Класификацијата на слики снимени од далечина (Remote Sensing (RS)) во последно време привлекува големо внимание поради примената во разрешување на различни задачи, вклучувајќи ја армијата, мониторинг на животната средина, надзор, обезбедување и детекција на геопросторни објекти. Воведувањето на конволуциски невронски мрежи (CNN) и други техники на „длабоко“ учење придонело до огромно подобрување на точноста на класификацијата на сцени снимени од воздух.

Во нашето истражување, користиме пренос на учење од претходно тренирани длабоки конволуциски невронски мрежи (CNN) во рамки на класификацијата на слики снимени (набљудувани) од далечина. Првиот дел од методите за пренос на учење, екстракција на карактеристики, изведен е со употреба на претходно тренирани конволуциски мрежи кои екстрахираат длабоки карактеристики на сликите снимени од воздух од различни мрежни нивоа: нивото за редукција на просек (average pooling) или некое од претходните конволуциски нивоа. Потоа, применето е спојување на екстрахираните карактеристики од различни невронски мрежи, откако е изведена редукција на димензионалноста на големите вектори на карактеристики. Конечно, искористена е машина со вектори за поддршка (Support Vector Machine (SVM)) за класификација на споените карактеристики.

Во вториот дел од истражувањето, предлагаме примена на пренос на учење со фино подесување на претходно тренираните конволуциски невронски мрежи за класификација „од крај до крај“ (end-to-end) на слики снимени од воздух. За подесување на стапката на учење, усвоивме примена на линеарно опаѓање на стапката на учење, како и примена на циклични стапки на учење. Уште повеќе, со цел да се ублажи проблемот со претренирање (overfitting) на мрежните модели, применивме регуларизација со измазнување на лабели. Нашиот пристап изведува екстракција на карактеристики од фино подесените невронски мрежи и класификација на сликите снимени од далечина со модел на машина со вектори за поддршка со линерано јадро и јадро со радијални основни

функции (Radial Basis Function (RBF)).

Ние експериментиравме екстензивно со различни мрежни архитектури, со цел да се добијат оптимални резултати: усвоивме конволуциски мрежи кои содржат модули со различни димензии на филтрите (inception modules), како и мрежи кои содржат модули што учат остатоци од функции (residual modules). Конкурентноста на испитуваните техники беше евалуирана на четири реални податочни множества: UC Merced, WHU RS, AID и NWPU-RESISC45. Добиената точност на класификација покажува дека разгледуваните методи даваат конкурентни резултати споредено со други современи техники. Резултатите покажуваат дека предложените методи покажуваат точност на класификација до 98%, со што се надминуваат други современи техники.

Клучни зборови: набљудување од далечина; екстракција на карактеристики; фузија на карактеристики; фино подесување; циклични стапки на учење; измазнување на лабели.

AERIAL SCENE CLASSIFICATION WITH TRANSFER LEARNING METHODS FROM CONVOLUTIONAL NEURAL NETWORKS

Abstract: Scene classification relying on images is essential in many systems and applications related to remote sensing. Remote Sensing (RS) image classification has recently attracted great attention for its utilization in different tasks, including military, environmental monitoring, surveillance, security, and geospatial object detection. The introduction of convolutional neural networks (CNN) and other deep learning techniques contributed to vast improvements in the accuracy of image scene classification. In our research, we use transfer learning from pre-trained deep Convolutional Neural Networks (CNN) within remote sensing image classification. We performed the first part of the methods for transfer learning, the feature extraction, using pre-trained CNN that extracts deep features of aerial images from different network layers: the average pooling layer or some of the previous convolutional layers. Next, we applied feature concatenation on extracted features from various neural networks, after dimensionality reduction was performed on enormous feature vectors. Finally, we used the Support Vector Machine (SVM) for the classification of the concatenated features. In the second part of the research, we propose the adoption of transfer learning by fine-tuning pre-trained CNNs for end-to-end aerial image classification. To tune the learning rate hyperparameter, we employed a linear decay learning rate scheduler as well as cyclical learning rates. Moreover, to mitigate the overfitting problem of pre-trained models, we applied label smoothing regularization. Our approach performs feature extraction from the fine-tuned neural networks and remote sensing image classification with a Support Vector Machine (SVM) model with linear and Radial Basis Function (RBF) kernels.

We experimented extensively with different CNN architectures, to get optimal results: we adopted the inception-based CNNs, as well the residual-based networks. The competitiveness of the examined techniques was evaluated on four real-world datasets: UC Merced, WHU RS, AID and NWPU-RESISC45. The obtained classification accuracies demonstrate that the considered methods have competitive results compared to other cutting-edge techniques. The results show that the proposed methods exhibit classification accuracy of up to 98%, outperforming other state-of-the-art techniques.

Keywords: remote sensing; feature extraction; feature fusion; fine-tuning; cyclical learning rates; label smoothing

СОДРЖИНА

1. ВОВЕД

1.1 Вештачки невронски мрежи	7
1.2 Невронски мрежи со проследување нанапред (feedforward)	9
1.3 Алгоритам со повратно проследување (Backpropagation algorithm)	10
1.3.1 Прв случај: излезно ниво	12
1.3.2 Втор случај: тежината припаѓа на скриен слој	14
1.3.3 Варијанти на алгоритмот со повратно проследување	16
1.3.4 Момент-базиран алгоритам со повратно проследување	17
1.3.5 Нестеров (Nesterov) забрзување	18
1.4 Претренирање (overfitting) и регуларизација	19
1.4.1 $L2$ регуларизација	20
1.4.2 $L1$ регуларизација	22
1.4.3 Испуштање (dropout)	23
1.4.4 Вештачко проширување на множеството за тренирање	24
1.5 Избор на хиперпараметри на невронската мрежа	25
1.6 Алгоритми за учење со адаптивни стапки	27
1.7 Конволуциски невронски мрежи	29
1.7.1 Локални рецептивни полиња	30
1.7.2 Делење на тежините	31
1.7.3 Нивоа за активација и нормализација на мини групите	33
1.7.4. Нивоа за редукација	34
1.7.5 Пренос на учење	35
2. ПРЕГЛЕД НА ЛИТЕРАТУРАТА	
2.1 Значење на класификацијата на сцени снимени од воздух	38
2.2 Класификација на воздушни сцени со методи на ниско и средно ниво	39
2.3 Класификација на воздушни сцени со методи на „длабоко“ учење	42
3. ЦЕЛ НА ИСТРАЖУВАЊЕТО	47
4. МЕТОДИ НА ИСТРАЖУВАЧКАТА РАБОТА	51
4.1 ResNet	51
4.2 Inception	53

4.3 Xception	55
4.4 DenseNet	56
4.5 Класификатор со логистичка регресија (Logistic Regression Classifier (LRC))	58
4.6 Машина со вектори за поддршка (Support Vector Machine (SVM))	58
4.7 Намалување на димензионалноста	59
4.8 Линеарно опаѓање на стапката на учење	60
4.9 Циклични стапки на учење	61
4.10 Измазнување на лабели	63
4.11 UC Merced податочно множество	64
4.12 WHU RS податочно множество	66
4.13 AID податочно множество	67
4.14 NWPU-RESISC45 податочно множество	68
4.15 Експериментална поставка за предложениот метод со екстракција на карактеристики	71
4.16 Експериментална поставка за предложениот метод со фино подесување	74
4.17 Мерки за евалуација	76
5. РЕЗУЛТАТИ	
5.1 Резултати од методот со екстракција на карактеристики	
5.1.1 Класификација базирана на екстрахирани карактеристики од различни мрежни нивоа	78
5.1.2 Класификација базирана на фузија на карактеристики трансформирани со анализа на главните компоненти	84
5.2 Резултати од методот со фино подесување	
5.2.1 Класификација на AID податочното множество	99
5.2.2 Класификација на NWPU-RESISC45 податочното множество	117
6. ДИСКУСИЈА	128
6.1 Дискусија за резултатите добиени со методот со екстракција на карактеристики	128
6.2 Дискусија за резултатите добиени со методот со фино подесување	130
7. ЗАКЛУЧОК	134

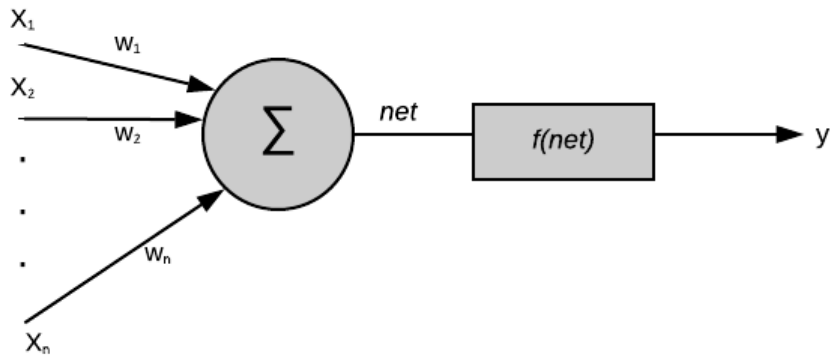
8. КОРИСТЕНА ЛИТЕРАТУРА (REFERENCES)	136
9. ДОДАТОК – користени кратенки	148

1. ВОВЕД

1.1 Вештачки невронски мрежи

Развојот на невро-компјутерите започнал во средината на 20-иот век со низа на биолошки откритија и нивно соодветно моделирање за компјутерска обработка. McCulloch et Pitts (1943) го дефинирале моделот на вештачки неврон (TLU-Threshold Logic Unit перцептрон) и покажале дека со него може да се пресметуваат логичките функции (И, ИЛИ и НЕ). Неколку години подоцна, Hebb (1949) ги објавил своите сознанија за работата и интеракцијата на биолошките неврони. Конкретно, тој заклучил дека ако два неврона се „палат“ заедно, тогаш доаѓа до метаболички промени со кои ефикасноста со која едниот неврон го побудува другиот расте со тек на времето. Со овие сознанија биле создадени услови за дефинирање на алгоритмот на учење на перцептронот. Тоа го направил Rosenblatt (1957, 1962) дефинирајќи го учењето на перцептронот (користејќи ја идејата на Hebb) како менување на јачината на врската меѓу невроните.

Биолошкиот неврон се состои од дендрити, тело на невронот и аксон. Во просек, секој неврон е поврзан со 10^4 други неврони преку дендритите и преку нив собира електрични импулси кои ги акумулира во телото. Откако ќе се акумулира одредена количина на електричество (грубо одредена со праг), невронот се пали – го проследува акумулираното електричество по својот аксон кон другите неврони и на тој начин се празни. Во таа смисла, дендритите претставуваат влезови преку кои невронот собира информации, телото на клетката ги обработува и на крајот генерира резултат кој се пренесува преку аксонот - излезот на невронот. Врз основа на овој поедноставен опис дефиниран е основниот модел на невронот кој е прикажан на сл.1.1. Моделот се состои од влезови x_1 до x_n („дендрити“), тежини w_1 до w_n кои одредуваат во која мерка секој од влезовите го побудува невронот, тело на невронот кое ја пресметува вкупната побуда (која ќе се означува со net) и преносна функција $f(net)$ (аксон на невронот) која ја обработува побудата и ја проследува на излез на невронот.



Слика 1.1 Основен модел на неврон

Figure 1.1 Plain model of neuron

Вкупната побуда net се пресметува според изразот:

$$net = w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n - \theta \quad (1.1)$$

каде што θ претставува праг на палење на невронот. За да не се третира посебно изразот за прагот во изразите за net , се дефинира дека невронот има уште еден „фиктивен“ влез x_0 на кој е доведена вредност 1, а прагот θ се заменува со тежина w_0 со што се добива:

$$net = w_0 \cdot x_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = \sum_{i=0}^n w_i \cdot x_i = \vec{w} \cdot \vec{x} \quad (1.2)$$

при што е $\vec{x} = (1, x_1, x_2, \dots, x_n)$ $(n + 1)$ - димензионален вектор кој претставува влез во невронот, а $\vec{w} = (w_0, w_1, \dots, w_n)$ $(n + 1)$ - димензионален вектор на тежините.

Преносната функција го моделира однесувањето на аксонот и вообичаено се користат неколку типични преносни функции. Преносната функција прикажана на сл.1.2 а) е функција на скок (step-function). Таа е дефинирана на следниот начин:

$$f(net) = \begin{cases} 0, & net \leq 0 \\ 1, & \text{во други случаи.} \end{cases} \quad (1.3)$$

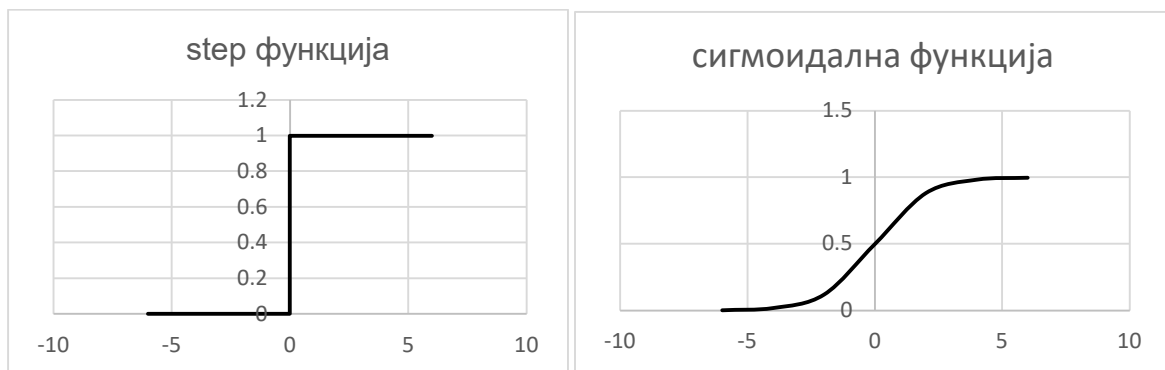
Оваа преносна функција ја користеле и McCulloch, Pitts (1943) при дефинирање на моделот на перцептронот. Друг вид на преносна функција е линеарна функција. Неврони со ваква преносна функција често се користат во излезните слоеви на невронските мрежи. Наместо линеарна функција, може да се користи и по делови линеарна функција. Проблемот со досега наведените типови на преносни функции е во тоа што кај посложените невронски мрежи кои

се составени од вакви елементи, не се знаело како да се дефинира алгоритмот на учење. Решение во облик на алгоритмот со повратно проследување (Backpropagation) се појавило дури со воведување на деривабилни преносни функции, чиј најпознат претставник е сигмоидална (или логистичка) преносна функција која е прикажана на сл.1.2 б). Таа е дефинирана со изразот:

$$f(net) = \frac{1}{1+e^{-net}}. \quad (1.4)$$

За оваа функција важи:

$$\frac{df(net)}{dnet} = f(net) \cdot (1 - f(net)). \quad (1.5)$$



а) Функција на скок

б) Сигмоидална функција

a) Step-function

b) Sigmoidal function

Слика 1.2 Типични преносни функции

Figure 1.2 Typical transfer functions

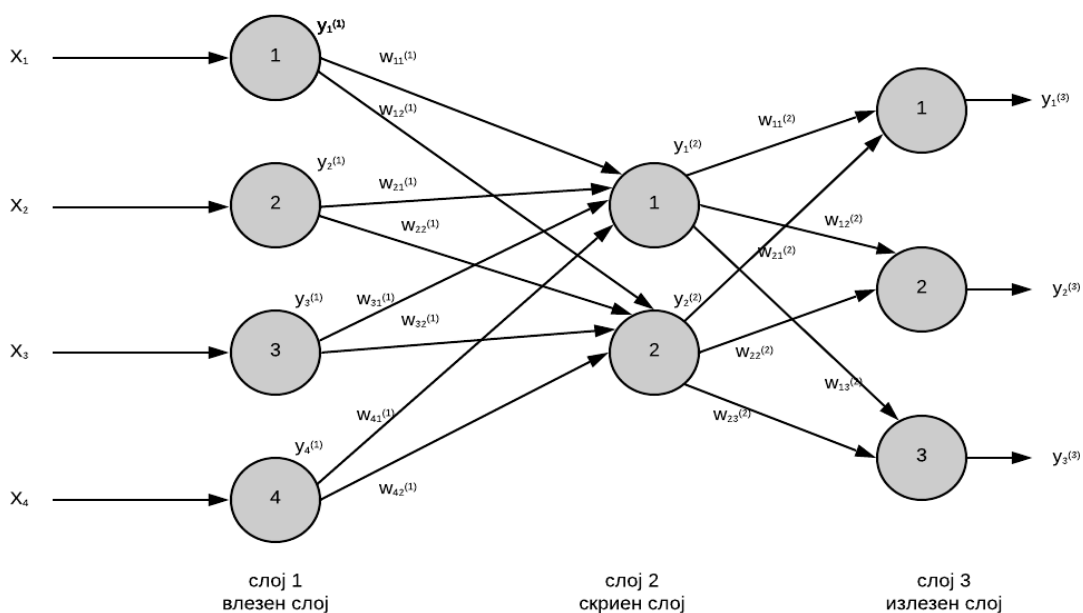
1.2 Невронски мрежи со проследување нанапред (feedforward)

Невронската мрежа со проследување нанапред (често позната и под називот повеќеслоен перцептрон, односно multilayer perceptron) е вид на невронска мрежа чија структура е прикажана на сл.1.3. Прикажаната невронска мрежа се состои од три слоја:

- 1-иот слој, односно влезниот слој се состои од неврони кои не вршат никаква функција, туку само ги пресликуваат доведените влезни податоци и ги прават достапни на остатокот на мрежата. Прикажаната мрежа на сликата има 4 неврони во влезниот слој и работи со 4-димензионални податоци. Влезот може да се запише како $\vec{x} = (x_1, x_2, x_3, x_4)$. За невроните од овој слој важи $y_i^{(1)} = x_i$.
- 2-иот слој, односно скриениот слој се состои од неврони кои на своите

влезови ги добиваат исклучиво вредностите од излезите на неврните од 1-иот слој. Според тоа, излезот на неврнот i од 1-иот слој на неврнот j од вториот слој влијае со тежина $w_{ij}^{(1)}$.

- 3-иот слој односно излезниот слој се состои од неврони кои на своите влезови ги добиваат единствено вредностите од излезите на неврните од 2-иот слој. На прикажаната слика има 3 неврони во излезниот слој, што значи дека прикажаната невронска мрежа генерира 3-димензионални излези, односно $\vec{y}^{(3)} = (y_1^{(3)}, y_2^{(3)}, y_3^{(3)})$.



Слика 1.3 Невронска мрежа со проследување нанапред

Figure 1.3 Feedforward neural network

Да биде невронската мрежа со проследување нанапред, нужен и доволен услов е да нема циклуси. Алгоритамот со повратно проследување за учење на невронски мрежи е применлив на поширока категорија на овој тип на мрежи.

1.3 Алгоритам со повратно проследување (Backpropagation)

Најпознатиот алгоритам за учење на невронски мрежи се изведува од воопштена слоевита невронска мрежа со проследување нанапред која има d влезови, m излези и за чие учење на располагање има N парови (\vec{x}_i, \vec{t}_i) каде што \vec{x}_i d -димензионален вектор кој го претставува i -тиот влезен примерок, а \vec{t}_i m -

димензионален вектор кој го претставува саканиот одзив на мрежата за разгледуваниот i -ти влезен примерок. Се претпоставува дека сите неврони имаат сигмоидална преносна функција и дека важи:

$$y = \frac{1}{1+e^{-net}} \quad (1.6)$$

каде што net е тежинска сума за набљудуваниот неврон. Овде се разгледува мрежа која има $k + 1$ слој: 1-иот слој е влезен, од 2-иот до k -тиот слој се скриени слоеви и $(k + 1)$ -иот слој е излезен слој. Невроните во секој слој се нумерирани на тој начин што $y_i^{(k)}$ означува излез на i -тиот неврон во k -тиот слој.

За да се измери квалитетот на невронската мрежа за потребите на постапката на учење, потребно е да се дефинира критериумска функција. Овде се дефинира критериумска функција еднаква на средното квадратно отстапување меѓу секој посакуван излез на мрежата и вистинските вредности кои мрежата ги генерира на излезот и тоа кумулативно за сите расположиви примероци. Бидејќи излезниот слој за набљудуваната мрежа е $(k + 1)$ -иот слој, тогаш важи:

$$E = \frac{1}{2N} \sum_{s=1}^N \sum_{o=0}^m (t_{s,o} - y_{s,o}^{(k+1)})^2 \quad (1.7)$$

при што индексот s се движи по сите примероци за учење, а индексот o се движи по сите излезни примероци. Ознаката $t_{s,o}$ претставува o -та димензија на саканиот излез кој му е придружен на s -тиот примерок за учење, додека ознаката $y_{s,o}^{(k+1)}$ е излезот на o -тиот неврон од излезниот слој (значи $(k + 1)$ -иот слој) кој го генерира мрежата кога на влезот и е донесен токму s -тиот примерок за учење.

Дефинираната функција E е функција од зададеното множество на примероци (кое е непроменливо за постапката за учење) и од вредноста на тежинските фактори, под претпоставка дека структурата на мрежата е исто така непроменлива. Во тој случај на вредноста на функцијата E може да се влијае само со промена на вредностите на тежинските фактори. Затоа, задачата на постапката на учење ќе биде да се пронајдат такви вредности на тежинските фактори за кои вредноста на функцијата ќе биде минимална. Значи, потребно е да се пресмета градиент од E кој го сочинуваат сите парцијални изводи од E по секој од тежинските фактори, т.е. потребно е да се пресмета:

$$\frac{\partial E}{\partial w_{ij}^{(k)}}. \quad (1.8)$$

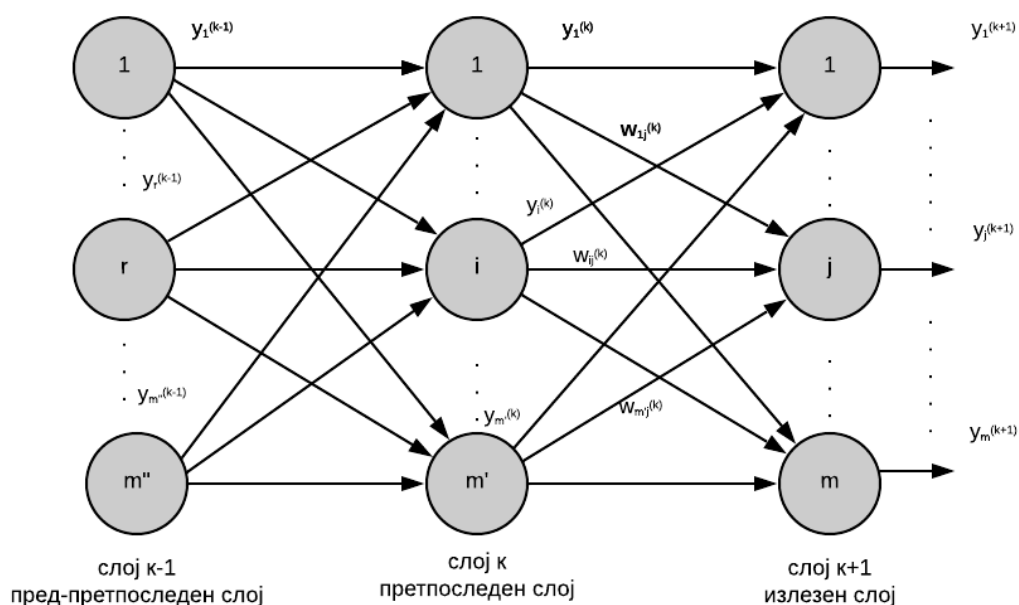
Тогаш, во согласност со градиентниот спус (gradient descent), може да се ажурираат тежините $w_{ij}^{(k)}$ на следниот начин:

$$w_{ij}^{(k)} \leftarrow w_{ij}^{(k)} - \psi \cdot \frac{\partial E}{\partial w_{ij}^{(k)}} \quad (1.9)$$

каде што ψ е мала позитивна константа.

1.3.1 Прв случај: излезно ниво

Најпрво ќе се разгледува случај во кој тежинскиот фактор припаѓа на неврон од излезниот слој. Таква состојба е прикажана на сликата 1.4.



Слика 1.4 Алгоритам со повратно проследување за неврон од излезниот слој

Figure 1.4 Backpropagation algorithm for output layer neuron

Потребно е да се пресмета $\frac{\partial E}{\partial w_{ij}^{(k)}}$. Може да се напише:

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = - \sum_{s=1}^N \sum_{o=1}^m (t_{s,o} - y_{s,o}^{(k+1)}) \cdot \frac{\partial y_{s,o}^{(k+1)}}{\partial w_{ij}^{(k)}} \quad (1.10)$$

Бидејќи тежината $w_{ij}^{(k)}$ ги спојува i -тиот неврон во k -тиот слој и j -тиот неврон во $k+1$ -иот слој, таа влијае само на излезот на j -тиот неврон во $(k+1)$ -иот слој. Затоа парцијалните изводи $\frac{\partial y_{s,o}^{(k+1)}}{\partial w_{ij}^{(k)}}$ во случај кога $o \neq j$ исчезнуваат и важи:

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = -\frac{1}{N} \sum_{s=1}^N (t_{s,j} - y_{s,j}^{(k+1)}) \cdot \frac{\partial y_{s,j}^{(k+1)}}{\partial w_{ij}^{(k)}}. \quad (1.11)$$

За пресметка на преостанатите парцијални деривации се применува правилото:

$$\frac{\partial y_{s,j}^{(k+1)}}{\partial w_{ij}^{(k)}} = \frac{\partial y_{s,j}^{(k+1)}}{\partial net_{s,j}^{(k+1)}} \cdot \frac{\partial net_{s,j}^{(k+1)}}{\partial w_{ij}^{(k)}}. \quad (1.12)$$

За сигмоидалната преносна функција важи:

$$\frac{\partial y_{s,j}^{(k+1)}}{\partial net_{s,j}^{(k+1)}} = y_{s,j}^{(k+1)} \cdot (1 - y_{s,j}^{(k+1)}). \quad (1.13)$$

Тежинската сума $net_{s,j}^{(k+1)}$ за неврон од $(k + 1)$ -иот слој се пресметува како:

$$net_{s,j}^{(k+1)} = w_{1j}^{(k)} \cdot y_{s,1}^{(k)} + w_{2j}^{(k)} \cdot y_{s,2}^{(k)} + \dots \dots \dots w_{ij}^{(k)} \cdot y_{s,i}^{(k)} + \dots \dots \dots \quad (1.14)$$

Поради особините на слоевитост на мрежата (нема циклични и латерални врски) ниту еден од излезите $y_{s,1}^{(k)}, y_{s,2}^{(k)} \dots \dots$ врз основа на кои се пресметува $net_{s,j}^{(k+1)}$ не зависат од тежината $w_{ij}^{(k)}$, туку се константи во однос на неа.

Парцијалниот извод на тежинската сума е:

$$\frac{\partial net_{s,j}^{(k+1)}}{\partial w_{ij}^{(k)}} = y_{s,i}^{(k)}. \quad (1.15)$$

Следи дека:

$$\frac{\partial y_{s,j}^{(k+1)}}{\partial w_{ij}^{(k)}} = y_{s,j}^{(k+1)} \cdot (1 - y_{s,j}^{(k+1)}) \cdot y_{s,i}^{(k)} \quad (1.16)$$

што со замена во изразот (1.11) дава:

$$\frac{\partial E}{\partial w_{ij}^{(k)}} = -\frac{1}{N} \sum_{s=1}^N \delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)} \quad (1.17)$$

при што е:

$$\delta_{s,j}^{(k+1)} = y_{s,j}^{(k+1)} \cdot (1 - y_{s,j}^{(k+1)}) \cdot (t_{s,j} - y_{s,j}^{(k+1)}). \quad (1.18)$$

Величината $\delta_{s,j}^{(k+1)}$ ја претставува грешката на j -иот излезен неврон за s -тиот примерок за учење. Правилото за ажурирање на тежините $w_{ij}^{(k)}$ гласи:

$$w_{ij}^{(k)} \leftarrow w_{ij}^{(k)} - \psi \cdot \frac{\partial E}{\partial w_{ij}^{(k)}} \quad (1.19)$$

$$\leftarrow w_{ij}^{(k)} - \psi \cdot \left(-\frac{1}{N} \sum_{s=1}^N \delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)} \right) \quad (1.20)$$

$$\leftarrow w_{ij}^{(k)} + \eta \cdot \left(\sum_{s=1}^N \delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)} \right) \quad (1.21)$$

каде што наместо производот $\psi \cdot \frac{1}{N}$ се користи нова константа η , којашто се

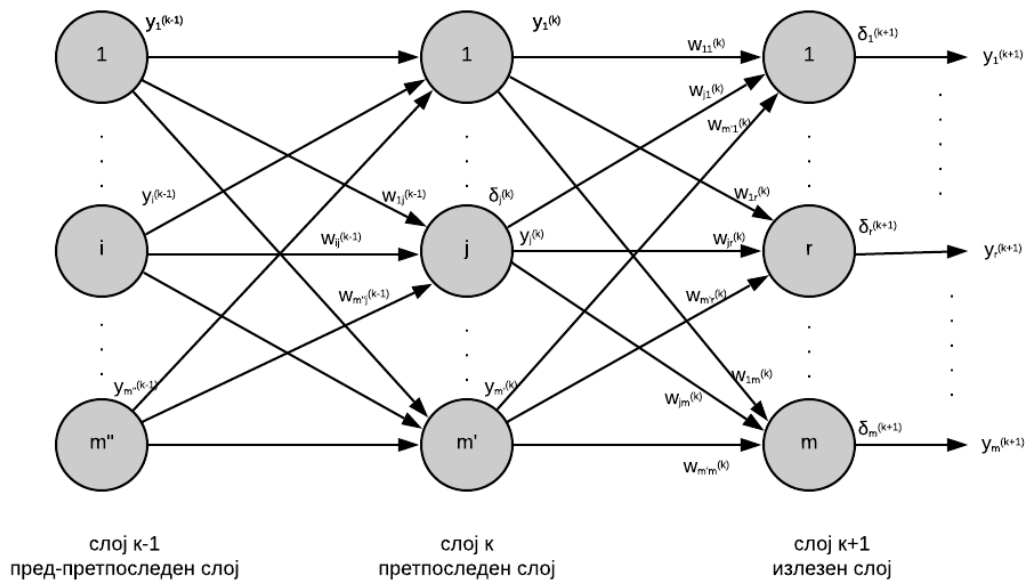
нарекува стапка на учење на невронската мрежа.

1.3.2 Втор случај: тежината припаѓа на скриен слој

Овој случај се разгледува на пример на тежина од неврон кој се наоѓа во последниот скриен слој, а потоа ќе се обопшти на произволен слој. Овој случај е прикажан на сл.1.5.

Потребно е да се пресмета парцијалниот извод на критериумската функција по разгледуваната тежина $w_{ij}^{(k-1)}$.

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(k-1)}} &= -\frac{1}{N} \sum_{s=1}^N \sum_{o=1}^m (t_{s,o} - y_{s,o}^{(k+1)}) \cdot \frac{\partial y_{s,o}^{(k+1)}}{\partial w_{ij}^{(k-1)}} \\ &= -\frac{1}{N} \sum_{s=1}^N \sum_{o=1}^m (t_{s,o} - y_{s,o}^{(k+1)}) \cdot \frac{\partial y_{s,o}^{(k+1)}}{\partial net_{s,o}^{(k+1)}} \cdot \frac{\partial net_{s,o}^{(k+1)}}{\partial w_{ij}^{(k-1)}} . \end{aligned} \quad (1.22)$$



Слика 1.5 Алгоритам со повратно проследување за неврон од скриен слој

Figure 1.5 Backpropagation algorithm for hidden layer neuron

Првиот парцијален извод е извод на сигмоидална преносна функција:

$$\frac{\partial y_{s,o}^{(k+1)}}{\partial net_{s,o}^{(k+1)}} = y_{s,o}^{(k+1)} \cdot (1 - y_{s,o}^{(k+1)}) . \quad (1.23)$$

За да се разреши вториот парцијален извод се фиксира o и се добива:

$$net_{s,o}^{(k+1)} = w_{1o} \cdot y_{s,1}^{(k)} + w_{2o} \cdot y_{s,2}^{(k)} + \dots \dots w_{jo} \cdot y_{s,j}^{(k)} + \dots \quad (1.24)$$

Поради тоа што мрежата е слоевита, тежината $w_{ij}^{(k-1)}$ која се наоѓа во

тежинската сума на j –иот неврон во k –тиот слој влијае само на неговиот излез, односно на $y_{s,j}^{(k)}$. Затоа $\forall o \neq j$ важи $\frac{\partial y_{s,o}^{(k)}}{\partial w_{ij}^{(k-1)}} = 0$, бидејќи промената на таа тежина

нема никакво влијание на тие излези. Бидејќи само $\frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}} \neq 0$, следува:

$$\frac{\partial net_{s,o}^{(k+1)}}{\partial w_{ij}^{(k-1)}} = w_{jo} \cdot \frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}}. \quad (1.25)$$

Со комбинирање на овие резултати следува:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^{(k-1)}} &= -\frac{1}{N} \sum_{s=1}^N \left[\frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}} \sum_{o=1}^m (t_{s,o} - y_{s,o}^{(k+1)}) \cdot y_{s,o}^{(k+1)} \cdot (1 - y_{s,o}^{(k+1)}) \cdot w_{jo} \right] \\ &= -\frac{1}{N} \sum_{s=1}^N \left[\frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}} \sum_{o=1}^m \delta_{s,o}^{(k+1)} \cdot w_{jo} \right]. \end{aligned} \quad (1.26)$$

Преостанатиот парцијален извод $\frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}}$ лесно се разрешува бидејќи тежината $w_{ij}^{(k-1)}$ е всушност тежина на невронот чиј излез е $y_{s,j}^{(k)}$. Во развиена форма изразот се пишува:

$$\begin{aligned} \frac{\partial y_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}} &= \frac{\partial y_{s,j}^{(k)}}{\partial net_{s,j}^{(k)}} \cdot \frac{\partial net_{s,j}^{(k)}}{\partial w_{ij}^{(k-1)}} \\ &= y_{s,j}^{(k)} \cdot (1 - y_{s,j}^{(k)}) \cdot y_{s,i}^{(k-1)} \end{aligned} \quad (1.27)$$

Со замена следува:

$$\frac{\partial E}{\partial w_{ij}^{(k-1)}} = -\frac{1}{N} \sum_{s=1}^N [y_{s,i}^{(k-1)} \cdot \delta_{s,j}^{(k)}] \quad (1.28)$$

при што грешката на скриениот неврон j во слојот k за примерокот s е дефинирана како извод на неговата преносна функција помножена со тежинската сума на грешките на невроните од следниот слој, односно:

$$\delta_{s,j}^{(k)} = y_{s,j}^{(k)} \cdot (1 - y_{s,j}^{(k)}) \cdot \sum_{o=1}^m \delta_{s,o}^{(k+1)} \cdot w_{jo}. \quad (1.29)$$

Ажурирањето на тежините $w_{ij}^{(k-1)}$ се одвива на следниот начин:

$$w_{ij}^{(k-1)} \leftarrow w_{ij}^{(k-1)} - \psi \cdot \frac{\partial E}{\partial w_{ij}^{(k-1)}} \quad (1.30)$$

$$\leftarrow w_{ij}^{(k-1)} - \psi \cdot \left(-\frac{1}{N} \sum_{s=1}^N [y_{s,i}^{(k-1)} \cdot \delta_{s,j}^{(k)}] \right) \quad (1.31)$$

$$\leftarrow w_{ij}^{(k-1)} + \eta \cdot \left(\sum_{s=1}^N [\delta_{s,j}^{(k)} \cdot y_{s,i}^{(k-1)}] \right) \quad (1.32)$$

каде што наместо производот $\psi \cdot \frac{1}{N}$ е воведена нова константа η , којашто се

нарекува стапка на учење на невронската мрежа.

Ако наместо неврон од последниот скриен слој, се разгледува неврон од некој од претходните слоеви, се доаѓа до структурно истите формули: за секој неврон од скриениот слој l грешката се пресметува како извод на неговата преносна функција помножена со тежинската сума на грешките на невроните од следниот слој на кој што разгледуваниот неврон е директно споен.

1.3.3 Варијанти на алгоритмот со повратно проследување

Изразот (1.21) за ажурирање на тежинските фактори се добива со директна примена на методот на градиентен спус на минимизација на средната квадратна грешка на целокупното множество на примероци за учење. Во тој израз сумата $\sum_{s=1}^N \delta_{s,j}^{(k+1)}$ произлегува од изразот за точната вредност на градиентот на критериумската функција во точка која е одредена со моменталните вредности на тежинските фактори. Алгоритмот за учење кој е остварен на ваков начин спаѓа во групата на алгоритми за групно учење (batch learning algorithm), кај кои учењето се остварува дури по презентација на сите примероци за учење. Во таа смисла, една епоха е еднаква на една итерација на вака изведениот алгоритам. Проблемот кај оваа изведба лежи во тоа што функцијата која се минимизира е високо мултимодална и градиентот во секоја точка секогаш покажува во насока на најблискиот локален минимум, но не во насока на глобалниот минимум. Поради ова, ваквата изведба на алгоритмот за учење покажува многу мала отпорност на заглавување во локалните минимума.

За да се реши овој проблем, можно е да се направи модификација на алгоритмот, така што наместо вистински вредности на градиентот во секоја точка, се прави проценка на градиентот само врз основа на еден примерок и потоа веднаш се прави корекција на тежините само врз основа на оваа проценка. Ваквата изведба е позната под името стохастички алгоритам со повратно проследување (Backpropagation) или стохастички градиентен спус (Stochastic Gradient Descent – SGD). Тоа се постигнува кога наместо сумата:

$$\sum_{s=1}^N \delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)} \quad (1.33)$$

во s –тата итерација се користи апроксимација на градиентот:

$$\delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)} \quad (1.34)$$

и корекција се прави само врз основа на таа апроксимација:

$$\forall s \in [1, \dots, N] \quad w_{ij}^{(k)} \leftarrow w_{ij}^{(k)} + \eta \cdot \delta_{s,j}^{(k+1)} \cdot y_{s,i}^{(k)}. \quad (1.35)$$

Со ова се постигнува алгоритмот да учи по секој презентирани примерок. Таква варијанта на алгоритмот припаѓа на семејството на алгоритми со поединечно учење (online learning algorithm). Кај оваа изведба алгоритмот прави една итерација со презентација точно на еден примерок и со корегирање на тежините, додека низата од N итерации кога на мрежата ќе ѝ се презентираат сите примероци се нарекува епоха. Постои можност невронската мрежа да прави итерација со презентирање на мини група од примероци (mini-batch) и корекцијата на тежините да ја прави за секоја мини група. За разлика од класичниот алгоритам со повратно проследување, стохастичкиот алгоритам со повратно проследување покажува зголемена отпорност кон заглавување во локални минимуми и поради тоа е најчесто користен алгоритам.

1.3.4 Момент-базиран алгоритам со повратно проследување

Момент-базираниот алгоритам со повратно проследување (или градиентен спуст) вклучува информација не само за градиентот, туку и информација за тоа како градиентот се менува. За да се разбере техниката на моментот (momentum), треба да си ја предочиме добро познатата слика на градиентен спуст, а тоа е топче кое се тркала надолу во долината.

Техниката на моментот го модифицира градиентниот спуст на два начина, со што го прави посличен со физичката слика. Прво, го воведува поимот за „брзина“ за параметрите кои се оптимизираат. Градиентот делува со цел да ја промени брзината, но (не директно) и „положбата“, исто како што и физичките сили ја менуваат брзината, а само индиректно положбата. Второ, момент-базираниот метод вклучува и член за триење, кој има тенденција постепено да ја намалува брзината. Променливите за брзината се $v = v_1, v_2, \dots$, по една за секоја тежина w_j . Правилото за промена на градиентниот спуст е дадено со:

$$v \leftarrow v' = \mu v - \eta \frac{\partial E}{\partial w} \quad (1.36)$$

$$w \leftarrow w' = w + v' \quad (1.37)$$

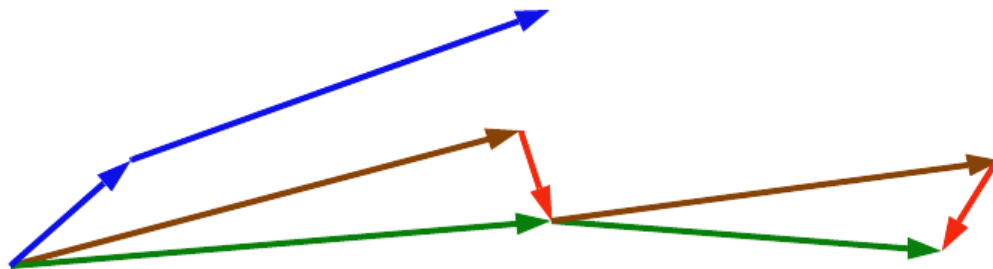
Членот за моментот треба да го зголеми интензитетот на ажурирањата за димензиите чии што градиенти ги менуваат насоките, Qian (1999) и Ruder (2016). Во релациите (1.36) и (1.37) μ е хиперпараметар кој ја контролира количината на триење во системот. Вредноста на μ која се движи меѓу 0 и 1 може да го забрза

учењето на мрежата, но без да се предизвика надминување (overshooting) на минимумите на критериумската функција.

Хиперпараметарот μ се вика момент коефициент. Тој е различен од поимот за момент во физиката и е многу поблизок до поимот за триење. Момент-базираната техника често се користи и вообичаено го забрзува тренирањето на невронската мрежа.

1.3.5 Нестеров (Nesterov) забрзување

Нестеров забрзувањето (Nesterov acceleration) е тесно поврзано со моментот кај стохастичкиот градиентен спуст. Ако моментот премногу се зголеми, тогаш можно е да се надмине локалниот минимум и да се продолжи со тренирање. Затоа, добро е да се препознае моментот кога треба да се забави со учењето на невронската мрежа, што е место каде што се воведува Нестеров забрзан градиент, Nesterov (1983). Нестеров забрзувањето може да се дефинира како корективно ажурирање на моментот, коешто ни овозможува да добиеме приближна идеја за тоа каде би биле мрежните параметри по ажурирањето. Визуелизација на Нестеров забрзувањето може да се види на сл.1.7, Hinton (2020).



Слика 1.7 Графички приказ на Нестеров забрзувањето

Figure 1.7 A graphical depiction of Nesterov acceleration

Со употреба на стандарден момент, прво се пресметува градиентот (малиот син вектор) и потоа се прави голем скок во насока на градиентот (големиот син вектор). Под Нестеров забрзувањето најпрво се прави голем скок во насока на претходниот градиент (кафеав вектор), се мери градиентот, а потоа се прави корекција (црвен вектор) – зелениот вектор е финално коригирано

ажурирање со Нестеров забрзување, Ruder (2016). Нестеров забрзувањето го проучувале и Bengio (2012) и Sutskever (2013).

1.4 Претренирање (overfitting) и регуларизација

Моделите со голем број на слободни параметри можат да опишат неверојатно широк опсег на феномени. Дури и ако таквиот модел се согласува добро со податоците кои се на располагање, тоа не го прави добар модел. Тоа можеби само значи дека постои доволно слобода во моделот со што може да се опише скоро секое податчно множество со определена големина, без да се има вистински увид во основната појава. Кога се случува ова, моделот ќе работи добро со постоечките податоци, но нема да успее да генерализира во нови ситуации. Вистински тест за моделот е неговата способност да прави предвидувања во ситуации на коишто претходно не бил изложен.

Невронските мрежи имаат слободни параметри кои се од редот на 100,000 параметри. Постоечките длабоки невронски мрежи понекогаш содржат милиони или дури билиони параметри. Ова е особено точно кај модерните мрежи кои често имаат многу голем број на тежини. Многу често невронските мрежи по одреден процес на тренирање не успеваат да генерализираат со очекувана точност на одреден дел од податочното множество (на податоците за тестирање). Во тој случај мрежата е претренирана (overtraining, overfitting) на делот од податочното множество предвиден за тренирање. Понатамошното тренирање (учење) на невронската мрежа не е корисно. Претренирањето е голем проблем кај невронските мрежи. За да може невронските мрежи ефективно да се тренираат, потребно е да се најде начин да се детектира претренирањето. Еден можен пристап е да се следи постигнатата точност на множеството за тестирање во текот на учење на мрежата. Ако се забележи дека точноста повеќе не се подобрува, потребно е да се стопира тренирањето.

Еден од најдобрите начини да се намали претренирањето е да се зголеми обемот на податоците за тренирање. Со доволно податоци за тренирање тешко е дури и големите мрежи да се претренираат. Меѓутоа, постапката на собирање податоци за тренирање може да биде скапа, а ова не е секогаш практична опција. Друг можен пристап е да се редуцира големината на невронската мрежа. Спротивно на ова, големите мрежи имаат потенцијал да бидат помоќни од малите мрежи, така што оваа опција неволно се усвојува од научната заедница,

Nielsen (2015). Постојат и други техники кои може да го намалат претренирањето, дури и во случаи кога структурата на мрежата и податочното множество за тренирање се фиксни. Тие се познати како регуларизациони техники. Меѓу најпознатите регуларизациони техники спаѓаат: $L2$ регуларизација, позната и како пропаѓање на тежините (weight decay), $L1$ регуларизација, испуштање (dropout) и вештачко проширување на податочното множество за тренирање (data augmentation).

1.4.1 $L2$ регуларизација

Идејата која лежи во основата на $L2$ регуларизацијата е да се додаде дополнителен член кај критериумската функција, т.н. член за регуларизација. Членот за регуларизација се воведува кај крос-ентропијата како критериумска функција и кај квадратната критериумска функција како што е прикажано подолу.

За еден неврон со сигмоидална (логистичка) преносна функција, крос-ентропијата како критериумска функција се дефинира со:

$$E = -\frac{1}{N} \sum_x [t \ln y + (1 - t) \ln(1 - y)] \quad (1.38)$$

каде што N е вкупниот број на примероци за тренирање, сумирањето се врши по сите влезови за тренирање x , со y е означен излезот на невронот, додека t е соодветниот посакуван излез. Лесно се генерализира крос-ентропијата на повеќеневронска мрежа со неколку слоеви. Нека $y = y_1, y_2 \dots \dots$ се вистинските излезни вредности, додека $t_1^{(k+1)}, t_2^{(k+1)} \dots \dots$ се посакуваните вредности на излезните неврони, т.е. на невроните од последното ниво. Тогаш крос-ентропијата се дефинира како:

$$E = -\frac{1}{N} \sum_x \sum_j [t_j \ln y_j^{(k+1)} + (1 - t_j) \cdot \ln(1 - y_j^{(k+1)})]. \quad (1.39)$$

Изразите (1.38) и (1.39) се идентични, освен што во изразот (1.39) сумирањето се врши по сите излезни неврони (\sum_j).

Регулизираната крос-ентропија е дадена со изразот:

$$E = -\frac{1}{N} \sum_{x,j} [t_j \ln y_j^{(k+1)} + (1 - t_j) \cdot \ln(1 - y_j^{(k+1)})] + \frac{\lambda}{2N} \sum_w w^2. \quad (1.40)$$

Првиот член е вообичаениот израз за крос-ентропија. Но додаден е и втор член, имено сумата на квадратите на сите тежини во мрежата. Овој член е помножен со $\frac{\lambda}{2N}$, каде што $\lambda > 0$ е познат како регуларизационен параметар, а N е големината на множеството за тренирање.

Можно е да се регуларизираат и други критериумски функции, како квадратната критериумска функција. Тоа се прави на сличен начин:

$$E = \frac{1}{2N} \sum_x \|t - y^{(k+1)}\|^2 + \frac{\lambda}{2N} \sum_w w^2. \quad (1.41)$$

Во двата случаја регулираната критериумска функција може да се напише како:

$$E = E_0 + \frac{\lambda}{2N} \sum_w w^2 \quad (1.42)$$

Интуитивно, ефектот на регуларизацијата е да се направи мрежата да преферира да учи мали тежини, додека другите нешта се исти. Големи тежини ќе се дозволуваат само ако тие значајно го подобруваат првиот дел од критериумската функција. Регуларизацијата може да се гледа како компромис меѓу изнаоѓањето на мали тежини и минимизирање на оригиналната критериумска функција. Релативната важност на двата елемента на компромисот зависи од вредноста на λ : кога λ е мало се преферира да се минимизира оригиналната критериумска функција, но кога λ е големо се преферираат мали тежини. Правилото за учење на тежините станува:

$$w \leftarrow w - \eta \frac{\partial E_0}{\partial w} - \frac{\eta \lambda}{N} w = \quad (1.43)$$

$$= \left(1 - \frac{\eta \lambda}{N}\right) w - \eta \frac{\partial E_0}{\partial w} \quad (1.44)$$

каде што η е стапка на учење на невронската мрежа. Ова е всушност истото правило за учење како кај градиентен спуст, со таа разлика што тежината прво се множи со $\left(1 - \frac{\eta \lambda}{N}\right)$. Оваа постапка понекогаш се нарекува и пропаѓање на тежините (weight decay), бидејќи ги прави тежините помали. На прв поглед изгледа дека тежините константно се намалуваат кон нула. Но тоа не е точно, бидејќи другиот член може да придонесува за зголемување на тежините, ако на тој начин се намалува нерегулираната критериумска функција.

Релациите дадени погоре се однесуваат на градиентен спуст. Во случај на стохастички градиентен спуст, исто како и кај нерегулираниот градиентен спуст, може да се направи проценка на $\frac{\partial E}{\partial w}$ со усреднување по сите m примероци за тренирање од мини групата (mini-batch). Регулираното правило за учење за стохастичкиот градиентен спуст станува:

$$w \leftarrow \left(1 - \frac{\eta \lambda}{N}\right) w - \frac{\eta}{m} \sum_x \frac{\partial E_x}{\partial w} \quad (1.45)$$

каде што сумирањето се врши за сите примероци за тренирање x од мини групата, а E_x е нерегуларизирана критериумска функција за секој примерок за тренирање. Изразот (1.45) е идентичен со вообичаеното правило за стохастички градиентен спуст, освен во делот за факторот за пропаѓање на тежините $1 - \frac{\eta\lambda}{N}$.

Регуларизацијата е начин да се намали претренирањето и да се зголеми точноста на класификација. Меѓутоа, ова не е единствената придобивка. При нерегуларизирано тренирање на невронската мрежа на алгоритмот за учење му е тешко правилно да го истражи просторот на тежините и затоа повремено заглавува во локалните минимума на критериумската функција. При регуларизирано тренирање на мрежата резултатите полесно се реплицираат, Nielsen (2015).

1.4.2 $L1$ регуларизација

Кај овој пристап се модифицира нерегуларизираната критериумска функција со додавање на сумата на апсолутните вредности на тежините:

$$E = E_0 + \frac{\lambda}{N} |w|. \quad (1.46)$$

Ова е слично со $L2$ регуларизацијата, „казнувајќи“ ги големите тежини и тежнеејќи кон тоа мрежата да преферира мали тежини. Членот за $L1$ регуларизација не е ист со членот за $L2$ регуларизација, па следствено на тоа не треба да се очекува потполно исто однесување на мрежата. Ако се пресметаат парцијалните деривации на критериумската функција се добива:

$$\frac{\partial E}{\partial w} = \frac{\partial E_0}{\partial w} + \frac{\lambda}{N} \text{sgn}(w) \quad (1.47)$$

каде што $\text{sgn}(w)$ е знакот на w , кој е $+1$ ако w е позитивна и -1 ако w е негативна.

Со употреба на овој израз може да се модифицира алгоритмот за учење да изведува стохастички градиентен спуст со користење на $L1$ регуларизација.

Правилото за промена на тежините кај $L1$ регуларизирана мрежа е:

$$w \leftarrow w - \frac{\eta\lambda}{N} \text{sgn}(w) - \eta \frac{\partial E_0}{\partial w} \quad (1.48)$$

каде што, како и вообичаено може да се направи проценка на $\frac{\partial E_0}{\partial w}$ со користење на усреднување во мини-групата. Ако се споредат релациите (1.44) и (1.48), кај $L1$ регуларизацијата тежините се намалуваат за константен износ кон 0. Кај $L2$ регуларизацијата тежините се намалуваат за износ кој е пропорционален на w . Кога одредена тежина има голема магнитуда $|w|$, $L1$ регуларизацијата ја

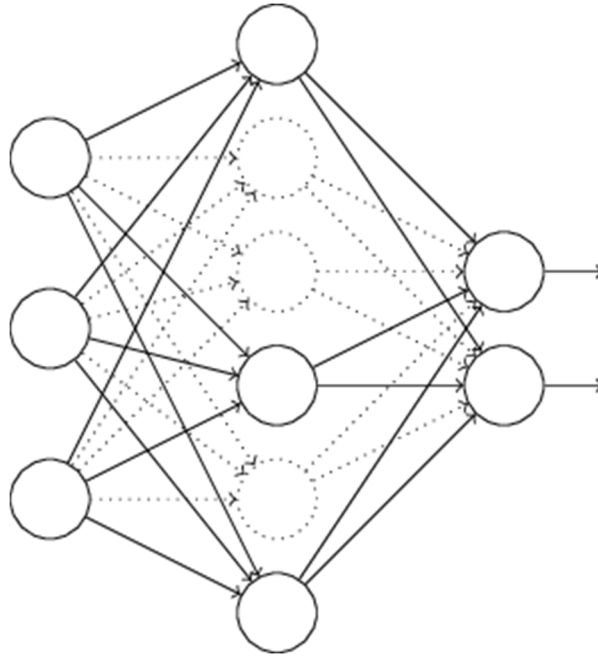
намалува тежината многу помалку од $L2$ регуларизацијата. Спротивно, кога $|w|$ е мала, $L1$ регуларизацијата ја намалува тежината многу повеќе од $L2$ регуларизацијата.

1.4.3 Испуштање (dropout)

Испуштањето е радикално различна техника за регуларизација. За разлика од $L1$ и $L2$ регуларизацијата, испуштањето не се базира на промена на критериумската функција, тука на промена на самата мрежа.

Да претпоставиме дека се обидуваме да тренираме невронска мрежа и дека имаме примерок x на влезот и соодветно посакуван примерок t на излезот. Вообичаено, тренирањето се изведува со пропагирање нанапред на x и потоа грешката се пропагира наназад за да се одреди придонесот кон градиентот. Со испуштањето овој процес е модифициран. Се почнува со случајно (и привремено) бришење на половина од скриените неврони во мрежата, додека влезните и излезните неврони се оставаат непроменети. По оваа операција мрежата добива изглед прикажан со полните линии од сл.1.7. Треба да се нагласи дека испуштените неврони, т.е. невроните кои беа привремено избришани, сè уште се присутни: влезот x се пропагира нанапред низ модифицираната мрежа, а потоа резултатот се пропагира наназад, исто низ модифицираната мрежа. Откако процесот ќе се спроведе на примероците од мини групата (mini-batch), се менуваат соодветните тежини. Постапката се повторува, прво со обновување на испуштените неврони, а потоа со случаен избор на ново подмножество на скриени неврони за бришење, проценка на градиентот за различна мини група и со ажурирање на тежините на мрежата.

Кога се испуштаат различни групи на неврони, тоа е како да се тренираат различни невронски мрежи. Процедурата за испуштање е еден вид усреднување на ефектите на голем број различни мрежи. Различните мрежи ќе бидат претренирани на различни начини, а ефектот на испуштањето би бил редуцирање на претренирањето. Испуштањето придонесува и во подобрување на перформансите на невронските мрежи. Техниката на испуштање е претставена од Hinton et al. (2012) со можност за примена во разрешување на различни задачи.



Слика 1.7 Испуштање кај повеќеслојна невронска мрежа
 Figure 1.7 Dropout for multi-layer neural network

1.4.4 Вештачко проширување на множеството за тренирање

Зголемувањето на множеството за тренирање е добро решение за претренирањето. Оваа постапка е скапа и не е секогаш можна. Меѓутоа, постои друга идеја која може да даде приближно ист ефект, а тоа е вештачко проширување на множеството за тренирање (data augmentation). Едно податочно множество, како MNIST множеството од слики, може да се прошири со мали ротации на сите MNIST слики за тренирање, а потоа да се користи проширеното множество за подобрување на преформансите на мрежата. MNIST податочното множество содржи десетици илјади скенирани слики од рачно напишани цифри, заедно со нивната точна класификација. Идејата за вештачко проширување на множеството за тренирање е широко распространета. Simard et al. (2003) примениле различни варијации на оваа идеја на MNIST. Тие го прошириле множеството за тренирање со користење не само на ротации, туку и со транслации и закосување на сликите. Исто така, експериментирале и со т.н. „еластични дисторзии“, посебен вид на дисторзии на сликата кои имаат за цел да ги имитираат случајните осцилации кои се среќаваат кај мускулите на дланката. Варијации на идејата за вештачко проширување на множеството за тренирање може да се користат за подобрување на перформансите на многу

задачи за учење, не само за распознавање на ракопис. Генерално, принципот се состои во тоа да се прошири множеството за тренирање со изведување на операции кои ги рефлектираат варијациите од реалноста.

1.5 Избор на хиперпараметри на невронската мрежа

Хиперпараметри при тренирање на невронската мрежа се: стапката на учење η , регуларизациониот параметар λ , големината на мини групата (mini-batch), момент коефициентот итн.

При избор на стапката на учење најпрво се одредува прагот за η при кој критериумската функција на податоците за тренирање веднаш почнува да опаѓа, наместо да осцилира или да се зголемува. На овој начин се проценува редот на величината на прагот за η . Ако вредноста на η треба да се користи многу епохи, тогаш вообичаено се избира помала вредност за η , на пр. два пати помала од прагот. Ваквиот избор дозволува мрежата да се тренира повеќе епохи, без да се забави премногу учењето. Често е поволно да се варира стапката на учење според т.н. распоред на стапката за учење (learning rate schedule). На почетокот на процесот на тренирање веројатно е дека тежините ќе бидат далеку од точни. Затоа е најдобро да се користи голема стапка на учење која ќе овозможи тежините брзо да се менуваат. Подоцна може да се намали стапката на учење, да се прават fino подесени прилагодувања на тежините на невронската мрежа.

Во однос на регуларизациониот параметар λ , најдобро е да се почне учењето на мрежата без регуларизација ($\lambda = 0$) и потоа да се одреди вредноста за η како што е опишано погоре. Откако ќе се одреди стапката на учење, се поминува на избор на соодветна вредност за λ . Може да се почне со $\lambda = 0.1$, а потоа да се зголемува или да се намалува за фактор 10 како што е потребно за да се подобрат перформансите на мрежата. Откако ќе се одреди добар ред на величината, потоа може fino да се подесува вредноста на λ . На крај, треба повторно да се оптимизира стапката на учење η .

При одредување на големината на мини групата (mini-batch) се разгледуваат повеќе аспекти. Прво се претпоставува дека ќе се изведува „онлајн“ (online) учење, т.е. дека ќе се користи мини група со големина 1. При користење на мини група која содржи само 1 примерок за тренирање, можно е да се јават значителни грешки во проценката на градиентот. Всушност, таквите

грешки не претставуваат голем проблем. Причината лежи во тоа што поединечните проценки на градиентот не мора да бидат супер прецизни. Сè што е потребно е проценката да биде доволно точна за да може критериумската функција да ја задржи тенденцијата на опаѓање. Изборот на големината на мини групата е компромис. Ако е премногу мала, не може да се искористат сите предности на матричните библиотеки кои се оптимизирани за брз хардвер. Ако е премногу голема, тогаш ажурирањето на тежините не се прави доволно често. Потребно е да се избере компромисна вредност која максимално ќе ја зголеми брзината на учење.

Рачната оптимизација на хиперпараметрите на невронската мрежа е добар начин за да се разбере како се однесува мрежата. Меѓутоа, вложени се напори да се автоматизира процесот. Вообичаена техника е мрежно пребарување (grid search), со која систематски преку мрежа се пребарува низ просторот на хиперпараметрите. Bergstra et al. (2012) направиле преглед на достигнувањата и ограничувањата на мрежното пребарување. Исто така, предложиле и пософистицирани пристапи. Snoek et al. (2012) користат Баесов пристап за автоматска оптимизација на хиперпараметрите.

Тешкотиите околу изборот на хиперпараметрите се зголемуваат и со фактот дека убедувањата како да се изберат хиперпараметрите се широко распространети, низ многу научни трудови и софтверски програми. Постојат повеќе трудови кои даваат препораки како да се постапува. Bengio (2012) дава некои практични препораки за употребата на алгоритмот со повратно проследување и градиентен спуст за тренирање на невронски мрежи, вклучувајќи длабоки архитектури. Меѓу другото, тој вклучува и детали како систематски да се пребаруваат хиперпараметрите. LeCun et al. (1998) имаат исто така добро истражување, но со постар датум.

1.6 Алгоритми за учење со адаптивни стапки

Кај стохастичкиот градиентен спуст стапката за учење е фиксна и ако е доволно мала, тогаш критериумската функција ќе опаѓа за време на тренирањето на мрежата. Меѓутоа, постојат и алгоритми за учење со адаптивни стапки за учење.

Еден таков метод е Адагард (Adagrad), прв пат претставен од Duchi et al. (2011). Адагард ја прилагодува стапката за учење на мрежните параметри. Поголеми промени се прават на параметри кои поретко се менуваат, додека помали промени се прават на параметри кои често се менуваат. Ажурирањето на тежините кај Адагард алгоритмот се прави со:

$$k \leftarrow k + \left(\frac{\partial E}{\partial w}\right)^2 \quad (1.49)$$

$$w \leftarrow w - \frac{\eta}{(\sqrt{k+\epsilon})} \frac{\partial E}{\partial w}. \quad (1.50)$$

Со променливата k се означува сумата на квадратите од градиентите по секој тежински фактор посебно и се ажурира на секоја мини група во процесот на тренирање. Тежините кои имаат често ажурирани големи градиенти во променливата k ќе го намалуваат размерот на промената, ефективно намалувајќи ја стапката за учење за тој параметар. Од друга страна, тежините кои имаат ретко ажурирани помали градиенти во променливата k ќе го зголемуваат размерот на промената, ефективно зголемувајќи ја стапката за учење за одредениот параметар. Примарната придобивка од Адагард е дека повеќе нема потреба рачно да се подесува стапката за учење – поголем дел од имплементациите на Адагард алгоритмот ја оставаат почетната стапка за учење и дозволуваат адаптивната природа на алгоритмот да ја подесува стапката за учење за секој параметар посебно.

Ададелта (Adadelata) алгоритмот бил предложен од Zeiler (2012). Ададелта може да се гледа како продолжение на Адагард кое бара да се редуцира монотоното опаѓање на стапката за учење кое е предизвикано од променливата k . Меѓутоа, Ададелта го ограничува ажурирањето на променливата со акумулирање само мал број на претходни градиенти – кога се имплементира, оваа операција резултира во пресметка на просечната вредност на пропаѓањето (decay) на сите претходни градиенти на квадрат. Ададелта може да се гледа како подобрување на Адагард, но многу блиску поврзан со RMSprop алгоритмот.

RMSprop алгоритмот за учење е развиен независно од Ададелта, не е објавен и е прикажан во предавањата од „Coursera” курсот на Hinton (2020). Слично како и Ададелта, RMSprop се обидува да ги поправи негативните ефекти на глобално акумулираната променлива k со нејзино претворање во

експоненцијално пондериран подвижен просек. Тежините на невронската мрежа со RMSprop алгоритмот се ажурираат со:

$$k \leftarrow \alpha \cdot k + (1 - \alpha) \cdot \left(\frac{\partial E}{\partial w}\right)^2 \quad (1.51)$$

$$w \leftarrow w - \frac{\eta}{(\sqrt{k} + \epsilon)} \cdot \frac{\partial E}{\partial w}. \quad (1.52)$$

Ако се погледнат релациите за RMSprop алгоритмот се забележува дека ажурирањето на тежините w е идентично со тоа кај Адагард – она што е важно е како се ажурира променливата k . Брзината на пропаѓањето, овде означена со α , е хиперпараметар кој типично има вредност 0.9. Може да се забележи дека претходните влезови во променливата k се пондерирани да бидат значајно помали од новите ажурирања. Овој „подвижен просек“ кај RMSprop дозволува од k да се испуштаат квадратите на старите градиенти и да се заменуваат со понови, „посвежи“. Во пракса RMSprop има тенденција да биде поефективен и од Адагард и од Ададелта кога се применува на различни длабоки невронски мрежи, Goodfellow et al. (2016). RMSprop конвергира значително побрзо од стохастичкиот градиентен спуст.

Алгоритмот за учење Адам (Adaptive Moment Estimation (Adam)) е предложен од Kingma et al. (2014) и е во суштина RMSprop со додаден момент:

$$m \leftarrow \beta_1 \cdot m + (1 - \beta_1) \cdot \frac{\partial E}{\partial w} \quad (1.53)$$

$$v \leftarrow \beta_2 \cdot v + (1 - \beta_2) \cdot \left(\frac{\partial E}{\partial w}\right)^2 \quad (1.54)$$

$$w \leftarrow w - \eta \cdot \frac{m}{\sqrt{v} + \epsilon}. \quad (1.55)$$

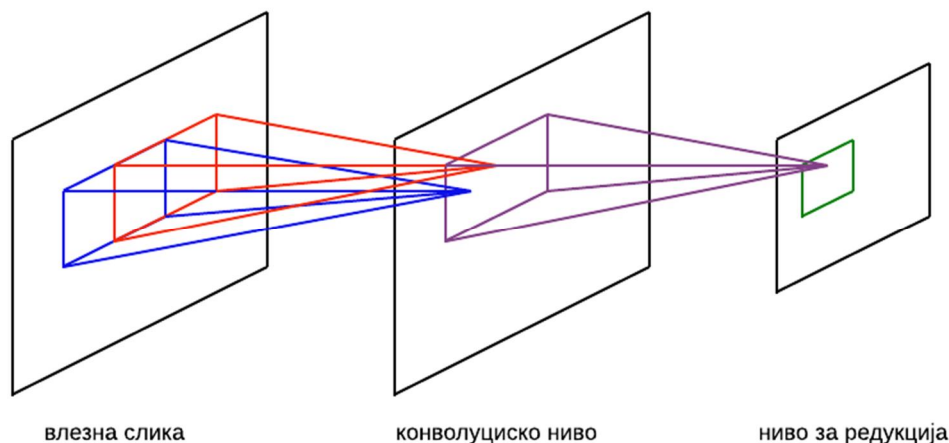
Вредностите за m и v се слични со моментот кај стохастичкиот градиентен спуст и се базираат на нивните соодветни претходни вредности од времето $t - 1$. Вредноста m претставува прв момент (средна вредност) на градиентот, додека v е втор момент (варијанса). Ажурирањето на тежините е скоро идентично со RMSprop, само што овде се користи „измазнета“ верзија на m , а не суров градиент $\frac{\partial E}{\partial w}$ – употребата на средна вредност води кон попосакувани ажурирања бидејќи можеме да ги измазниме суровите $\frac{\partial E}{\partial w}$ вредности кај кои е присутен шум. Адам има тенденција да работи подобро од RMSprop во многу ситуации.

Исто како што Адам е RMSprop со момент, Надам (Nadam) е RMSprop со Нестеров забрзување. Надам е предложен од Dozat (2015), докторант на Универзитетот „Стенфорд“.

1.7 Конволуциски невронски мрежи

Невронските мрежи со проследување нанапред (feedforward) кои се изградени од целосно поврзани слоеви не се користат за класификација на слики. Причината лежи во фактот што таквите мрежни архитектури не ја земаат предвид просторната структура на сликите. Тие ги третираат на ист начин влезните пиксели кои се оддалечени и оние кои се блиску. Таквите концепти на просторната структура треба да се изведат од податоците за тренирање. Конволуциските невронски мрежи имаат архитектура која се обидува да ја искористи просторната структура на сликите. Почетоците на конволуциските невронски мрежи датираат од 1970-ите години. Меѓутоа, основите на модерните конволуциски мрежи биле воспоставени од LeCun et al. (1998). Конволуциските невронски мрежи ги користат истите идеи како и мрежите со проследување нанапред: алгоритам со повратно проследување, градиентен спуст, регуларизација, нелинеарни активации, итн. Стохастичкиот градиентен спуст главно се користи во неговата изворна форма. Меѓутоа, потребно е да се направат извесни модификации, бидејќи основната изведба на алгоритмот е за невронски мрежи со проследување нанапред.

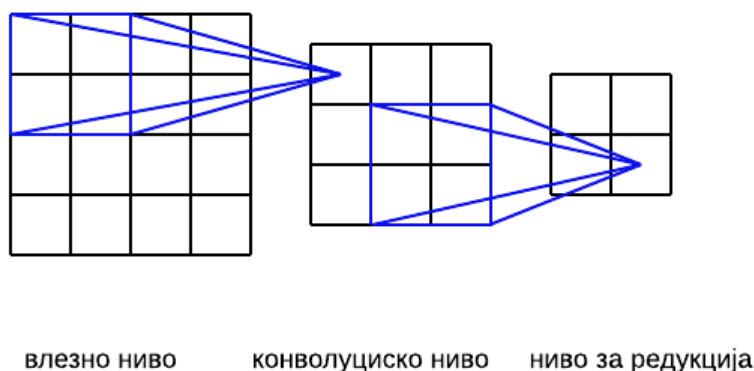
Често кај конволуциските мрежи поимите „вештачки неврон“ и „единица“ се употребуваат наизменично. Овие мрежи користат специјална архитектура која е посебно добро прилагодена за класификација на слики. Употребата на ваква архитектура овозможува конволуциските мрежи брзо да се тренираат. Тоа за возврат помага да се тренираат длабоки, многуслојни мрежи, кои се многу добри во класификацијата на слики. Денес, длабоки конволуциски мрежи или некоја блиска варијанта се користат во најголем дел на задачи за класификација на слики. Конволуциските невронски мрежи користат три базични идеи: локални рецептивни полиња, делење на тежините и редукција (pooling), сл.1.8 и сл.1.9.



Слика 1.8 Архитектура на конволуциска невронска мрежа
 Figure 1.8 The Architecture of CNN

1.7.1 Локални рецептивни полиња

Кај целосноповрзаните слоеви влезовите претставуваат вертикална линија од неврони. Кај конволуциската мрежа, влезовите се сфаќаат како $l \times m$ квадрати од неврони, чии што вредности кореспондираат со $l \times m$ интензитети на пикселите кои се користат како влезови. Влезните пиксели се поврзуваат со слој од скриени неврони. Но не се поврзува секој влезен пиксел со секој скриен неврон. Наместо тоа, конекции се прават во мали, локализирани региони од влезната слика. Тој регион од влезната слика се нарекува локално рецептивно поле за скриениот неврон. Тоа е мал прозорец на влезни пиксели. Секоја конекција учи одредена тежина, а скриениот неврон учи севкупен праг (bias). За скриениот неврон може да се смета дека учи да го анализира конкретното локално рецептивно поле. Потоа рецептивното поле се лизга низ целата влезна слика. Бројот на пиксели за кои се лизга рецептивното поле се нарекува должина на чекор (stride length). За секое рецептивно поле од влезното ниво постои различен скриен неврон во првото скриено ниво.



Слика 1.9 Конволуциско ниво –локални рецептивни полиња; ниво за редуција
 Figure 1.9 Convolutional layer – local receptive fields; pooling layer

1.7.2 Делење на тежините

Секој скриен неврон има праг и $l \times m$ тежини поврзани со неговото рецептивно поле. Истите тежини и праг ќе се користат за секој неврон од скриениот слој. За j, k –тиот скриен неврон излезот е:

$$\sigma(b + \sum_l \sum_m w_{lm} x_{j+l, k+m}). \quad (1.56)$$

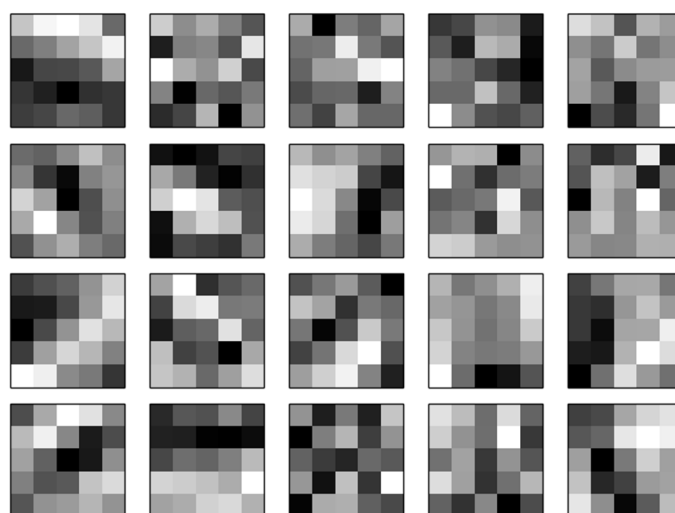
Тука σ е активациската (преносна) функција на невронот - можно е и сигмоидална функција, b е заедничката вредност за прагот, w_{lm} е низата на заедничките тежини и $x_{j,k}$ е влезната вредност на положбата j, k . Тоа значи дека сите неврони во скриеното ниво ја детектираат точно истата карактеристика (feature). На карактеристиката која се детектира од скриениот неврон треба да се гледа како на влезен примерок кој ќе предизвика активација на невронот: може да биде раб во сликата, или некоја друга форма, само на различни локации во влезната слика. Конволуциските мрежи се добро прилагодени на трансляциската инваријанса на сликите. Името конволуциски доаѓа од фактот дека операцијата во изразот (1.56) се нарекува конволуција.

Понекогаш мапата од влезниот слој кон скриениот слој се нарекува мапа на карактеристики (feature map). Тежините кои ја дефинираат мапата се нарекуваат споделени тежини, а прагот кој ја дефинира мапата на карактеристики се нарекува споделен праг. Споделените тежини и праг често се вели дека дефинираат јадро (kernel) или филтер. Голема предност на споделување на тежините и на праговите е дека тоа во голема мерка го намалува бројот на параметри во конволуциската мрежа. Употребата на

транслациската инваријанса кај конволуциското ниво ќе го намали бројот на параметри кои се потребни за да се добијат истите перформанси како кај целосноповрзаниот модел. Тоа резултира во побрзо тренирање на конволуцискиот модел, што во крајна линија помага да се изградат длабоки мрежи со употреба на конволуциски слоеви.

За да се изведува препознавање на слики потребно е да се користи повеќе од една мапа на карактеристики. Комплетното конволуциско ниво се состои од неколку различни мапи на карактеристики. Една од првите конволуциски мрежи, LeNet-5, користела 66 мапи на карактеристики, секоја поврзана со локално рецептивно поле со димензии 5×5 , за препознавање на MNIST цифри. На сл.1.10 се прикажани 20 различни мапи на карактеристики (или филтри, или јадра) од последното конволуциско ниво на невронска мрежа конструирана за препознавање на MNIST цифри.

Побелите блокови на сл.1.10 означуваат помала (типично понегативна) тежина и таму мапата на карактеристики помалку реагира на соодветните влезни пиксели. Потемните блокови означуваат поголема тежина и таму мапата на карактеристики реагира повеќе на соодветните влезни пиксели. Јасно е дека постои просторна структура која е повеќе од случајна: многу од карактеристиките на кои реагира конволуциското ниво имаат јасни подрегиони на светло или темно. Тоа покажува дека мрежата навистина учи нешта кои се поврзани со просторната структура.



Слика 1.10 Мапи на карактеристики од класификација на MNIST цифри

Figure 1.10 Feature maps from MNIST digits classification

1.7.3 Нивоа за активација и нормализација на мини групите

После секое конволуциско ниво во конволуциската невронска мрежа се применува нелинеарна активациска (преносна) функција, како „исправена линеарна единица“ (Rectified Linear Unit (ReLU)), „експоненцијална линеарна единица“ (Exponential Linear Unit (ELU)) или некоја од другите пропусни (Leaky) варијанти на „исправената линеарна единица“. Најчесто користена активација (преносна функција) е „исправена линеарна единица“. Активациските слоеви не се технички „слоевии“ (поради фактот дека параметрите/ тежините не се учат во рамки на ова ниво) и понекогаш не се прикажуваат во мрежните дијаграми, бидејќи се смета дека активацијата следи веднаш по конволуцијата.

Нормализацијата на мини групите е воведена од Ioffe et al. (2015). Словите за нормализација на мини групите се користат за нормализација на вредностите од одредена влезна единица пред да се проследат до следното ниво од мрежата. Ако сметаме дека x се влезовите од една мини група (mini-batch), тогаш нормализираните вредности \hat{x} се дадени со:

$$\hat{x}_i = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}}. \quad (1.57)$$

За време на тренирањето на мрежата μ_β и σ_β се пресметуваат за секоја мини група, при што:

$$\mu_\beta = \frac{1}{M} \sum_{i=1}^m x_i \quad \sigma_\beta^2 = \frac{1}{M} \sum_{i=1}^m (x_i - \mu_\beta)^2. \quad (1.58)$$

Вредноста за ε се зема мала и позитивна со цел да се избегне делење со 0. Излезите од нивото за нормализација на мини групите имаат средна вредност 0 и варијанса 1 (нула-центрирана). За време на тестирањето на конволуциската мрежа, вредностите μ_β и σ_β од мини групата се заменуваат со просечните вредности на μ_β и σ_β добиени во процесот на тренирање. Со ова се овозможува да се пропуштаат слики низ мрежата и да се добиваат точни предвидувања, без пристрасноста која се наметнува од μ_β и σ_β од последната мини група која минува низ мрежата за време на тренирањето.

Нормализацијата на мини групите се покажала како екстремно ефективна во намалувањето на бројот на епохи потребни за тренирање на конволуциските мрежи. Нормализацијата на мини групите помага и во „стабилизација“ на тренингот, дозволувајќи поширок опсег на стапки за учење и интензитети на

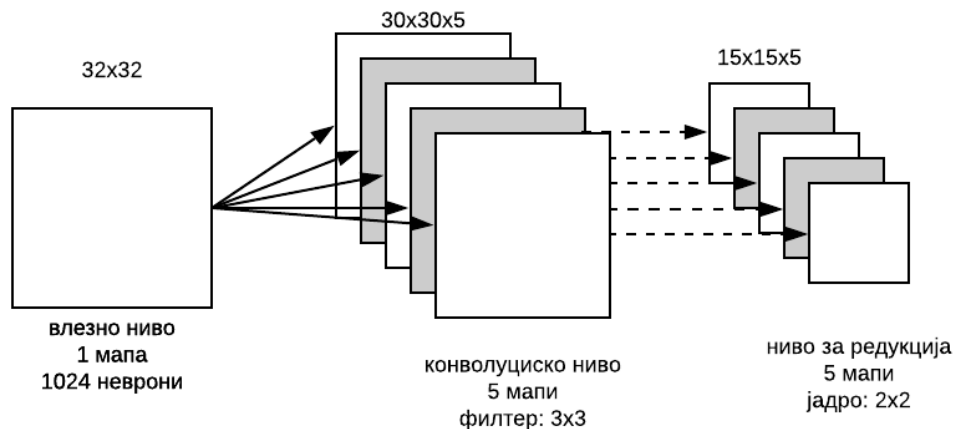
регуларизација. Примената на нормализација на мини групите во архитектурата на конволуциската мрежа помага да се превенира претренирањето (*overfitting*) и дозволува да се постигне повисока точност на класификација во помал број на епохи споредено со истата мрежна архитектура без ваква нормализација.

Според оригиналните истражувања на Ioffe et al. (2015), тие го позиционирале нивото за нормализација на мини групите пред активациското ниво. Нејасно е зошто Ioffe et al. (2015) предложиле позиционирање на нивото за нормализација на мини групите пред нивото за активација, меѓутоа понатамошните експерименти како кај Mishkin D. (2020) и искуствата од други експерти по длабоко учење прикажани во „Reddit community contributors“ (2020) потврдуваат дека поставувањето на слојот за нормализација на мини групите по нелинеарна активација резултира во повисока точност на класификација и помали загуби во скоро сите ситуации.

1.7.4. Нивоа за редукција

Конволуциските невронски мрежи покрај нивоата опишани во претходните секции содржат и нивоа за редукција (*pooling* нивоа). Редукциските нивоа се позиционираат по конволуциските нивоа. Функцијата на нивоата за редукција е да ја поедноставуваат информацијата на излезот од конволуциското ниво. Нивото за редукција го презема излезот од секоја мапа на карактеристики од конволуциското ниво и подготвува кондензирана мапа на карактеристики. На пример, секоја единица од нивото за редукција може да резимира регион од *lxm* неврони од претходното ниво. Една вообичаена постапка за редукција е редукцијата на максимум (*max-pooling*). Кај оваа редукција, единицата на излезот ја дава максималната активација од *lxm* влезниот регион. Конволуциското ниво содржи повеќе од една мапа на карактеристики и редукцијата на максимум се применува на секоја од нив поединечно, како што е прикажано на сл.1.11. Редукцијата на максимум е начин мрежата да пронајде дали одредена карактеристика се наоѓа некаде во регионот на сликата. Потоа, нивото ја исфрла точната информација за положбата. Голема придобивка е дека има многу помалку редуцирани карактеристики, па на овој начин се намалува бројот на потребните параметри во подоцнежните нивоа.

Редукцијата на максимум не е единствената техника за редукција. Друг вообичаен пристап е $L2$ редукција. Тука, наместо максималната активација од регионот од интерес, се зема квадратен корен од сумата на сите квадрати на активациите во разгледуваниот регион. Логиката е слична како кај редукцијата на максимум: $L2$ редукцијата е начин да се кондензира информацијата од конволуциското ниво. Во пракса двете техники широко се користат.



Слика 1.11 Ниво за редукција со јадро 2×2 за секоја мапа на карактеристики
Figure 1.11 Pooling layer with kernel 2×2 for each feature map

1.7.5 Пренос на учење

Целта на преносот на учење (transfer learning) е да се пренесе (трансферира) знаењето меѓу изворните и целните домени. При класификација на слики, главната придобивка од преносот на учење е да се надмине недостатокот на примероци за тренирање во одредени категории. Тоа се прави со адаптирање на класификатори тренирани за други категории. Невронската мрежа се здобива со знаење од податоците за тренирање. Знаењето се содржи во тежините на мрежата. Во случај кога има недостаток од податоци за тренирање, тренирањето на мрежата од почеток (со случајна распределба на тежините) не е корисно, ниту пак точноста е задоволителна. Во вакви ситуации, се задржуваат тежините на оригиналната мрежа и се пренесуваат научените карактеристики.

Преносот на учење досега успешно се применувал на класификација на расположение врз основа на текст (Wang et al. 2011), класификација на слики

(Duan et al. 2012; Zhu et al. 2011), класификација на човечка активност (Harel et al. 2011), класификација на софтверски дефекти (Nam et al. 2015), класификација на текст на повеќе јазици (Zhou et al. 2014). Pan et al. (2010) објавиле научен труд за пренос на учење. Оттогаш се објавени преку 700 академски трудови напишани на оваа тема. Тие го проучуваат преносот на учење на различни нивоа: со развој на нови алгоритми, со подобрување на постоечките алгоритми, како и со примена на алгоритмите во нови домени.

ImageNet податочното множество (Deng et al. 2009) е големо и разновидно. Тоа е причината поради која постоечките мрежи, претходно тренирани на ImageNet, покажуваат добра способност за класификација на слики надвор од податочното множество. Преносот на учење се изведува на три начини:

1. со екстракција на карактеристики – ова е состојба кога постоечкиот модел се користи како механизам за извлекување на карактеристики. Во овој случај излезното ниво се отстранува и потоа целата мрежа може да се користи како фиксен екстрактор на карактеристики;
2. со фино подесување – овој метод на пренос на учење се одликува со парцијално тренирање на претходно тренираната мрежа. Во овие случаи, тежините на пониските слоеви на мрежата се задржуваат непроменети, додека повисоките нивоа повторно се тренираат. Прашање е на проба колку нивоа да се задржат непроменети и колку нивоа повторно да се тренираат. Вообичаено финото подесување се прави со мали стапки на учење, неколку редови на големини помали од стапките на учење со кои е тренирана оригиналната мрежа;
3. со употреба на архитектурата на претходно тренираниот модел – во ова сценарио сè што користиме од претходно тренираниот модел е архитектурата. Сите тежини се иницијализираат случајно. Мрежата се тренира согласно податочното множество од интерес.

Методите кои се засноваат на екстракција на карактеристики и на фино подесување на невронските мрежи се користат во случај кога податочното множество кое треба да се класифицира е со мал обем. Во случај кога се користи методот со екстракција на карактеристики, излезните нивоа треба да се модифицираат и прилагодат на зададениот проблем. Кај сценариото со фино подесување одреден број на пониски нивоа (k на пример) се замрзнуваат.

Преостанатите ($L - k$) повисоки нивоа на претходно тренираниот мрежен модел се тренираат повторно. Во одредени случаи на fino подесување (кога размерот на податочното множество кое се класифицира го дозволува тоа) можно е да се тренира целата мрежа. Имено, се задржува архитектурата и иницијалните тежини на моделот. Потоа, невронската мрежа се тренира со тежините кои се иницијализирани како кај претходно тренираниот модел.

2. ПРЕГЛЕД НА ЛИТЕРАТУРАТА

2.1 Значење на класификацијата на сцени снимени од воздух

Еден од проблемите кои ги решава компјутерскиот вид е класификацијата на слики и истиот е темелно проучен во литературата. Постојат многу алгоритми кои го решаваат овој проблем. Класификацијата на сцени снимени од воздух е поголем предизвик поради фактот што објектите имаат случајна ротација и заднинската текстура е комплексна. Класификацијата на сцени снимени од воздух е една од главните задачи во разбирањето на сликите снимени од воздух. Во овој процес, на сцените снимени од воздух им се доделуваат семантички лабели, кои се утврдени врз основа на човековата интерпретација (Qayyum et al. 2017; Gan et al. 2016). Класификацијата на сцени е исто така позната и како класификација на слики снимени од далечина кога се работи за податочни множества добиени со набљудување од далечина (remote sensing). Овие два изрази се подеднакво застапени во литературата. Ваквата класификација нашла своја примена во многу области: на боишта, во контрола на сообраќајот, набљудување на катастрофи (Yang et al. 2015; Huang et al. 2015), надзор на аеродроми и заштита на авиони, набљудување на посеви во земјоделието и препознавање на промените на земјините покривки во инженерство на животната средина (Liang et al. 2016). Сликите снимени од далечина се композиција од информации за сцената и содржат голем број на карактеристики кои ја опишуваат текстурата. Композицијата на сцената кај сликите снимени од далечина е често сложена и поради тоа може да биде предизвикувачки да се добие семантичката лабела директно од сликите (Penatti et al. 2015; Luus et al. 2015). Ова е причината поради која се развиваат бројни техники за класификација на сцени.

Класификацијата на сликите снимени од далечина (Remote Sensing (RS)) е можна благодарение на достапноста на неколку податочни множества кои се собрани од платформите за набљудување на Земјата, како сателити, воздушни системи и беспилотни летала (UAV). Овие податочни множества бројат од неколку стотини до неколку илјади сцени и не се доволни за тренирање на моделите за „длабоко“ учење со случајна распределба на параметрите. Ефективноста на решенијата базирани на длабоки конволуциски невронски мрежи лежи во можноста да се изведе пренос на учење (знаење) од претходно тренираните мрежи (Bazi et al. 2019). Преносот на учење (знаење) за

класификација на слики може да се изведе на различни начини, вклучувајќи екстракција на карактеристики и фино подесување (Lu et al. 2020; Yosinski et al. 2014).

2.2 Класификација на воздушни сцени со методи на ниско и средно ниво

Методите за класификација на сцени со екстракција на карактеристики, генерално, може да припаѓаат во една од следните категории: методи кои користат карактеристики на сликите на ниско ниво (Lowe 2004; Risojevic et al. 2011, 2012, 2013; Manjunath et al. 1996; Yang et al. 2008; dos Santos et al. 2010; Swain et al. 1991; Ojala et al. 2002; Luo et al. 2008, 2009, 2013; Oliva et al. 2009; Chen C. et al. 2015), методи кои користат приказ на сликите на средно ниво (Yang et al. 2010, 2011; Chen et al. 2011; Scheirer et al. 2012; Sheng et al. 2012; Perronnin et al. 2010; Negret et al. 2014; Lazbenik et al. 2006; Chen S. et al. 2015; Blei et al. 2003; Stricker et al. 1995; Haralick et al. 1973; Kusumaningrum et al. 2014; Zhang et al. 2015; Bosch et al. 2006) и методи кои користат карактеристики на сликите на високо ниво (Russakovsky et al. 2015; Hu et al. 2015; Savic et al. 2003; J'egon et al. 2012; Nogueira et al. 2017; Xia et al. 2010, 2017; Jia et al. 2014; Szegedy et al. 2014; Cheng et al. 2017a).

Со методите кои користат визуелни карактеристики на ниско ниво, класификацијата на сцени се изведува со визуелни дескриптори на ниско ниво: спектрални, текстуални, структурни итн. Loewe 2004 ја моделира локалната флукуација на структурата кај сликите снимени од далечина со трансформација на карактеристики независно од размерот (Scale Invariant Feature Transform (SIFT)) како локален дескриптор на структурата. Други дескриптори ја користат статистичката и глобалната распределба на специфичните сликови карактеристики, како податоци за бојата (Swain et al. 1991) и текстурата (Risojevic et al. 2011; Manjunath et al. 1996). Yang et al. (2008) го користат IKONOS податчното множество за да ги споредат текстурните карактеристики добиени со трансформација на карактеристики независно од размерот (SIFT) и Габор (Gabor) текстурните карактеристики. Dos Santos et al. (2010) ги споредуваат различните дескриптори на боја и текстура како хистограми на бојата и дескриптори на локалните бинарни примероци (Local Binary Pattern (LBP)) (Ojala et al. 2002). Luo et al. (2013) користат шест различни типови на дескриптори:

трансформација на карактеристики независно од размерот, радиометриски карактеристики, матрица на истовремена појава на нивото на сиво (Grey Level Co-Occurrence Matrix (GLCM)), Гаусови „веивлет“ (Gaussian wavelet) карактеристики (Luo et al. 2008), карактеристики на формата (Luo et al. 2009) и Габор филтри и прават комбинации од нив во композитни карактеристики со различни просторни резолуции за класификација на сцени снимени од воздух. Oliva et al. (2001) користат GIST и дескриптори базирани на трансформација на карактеристики независно од размерот. Други дескриптори кои се користат во литературата се: дескриптор на разликите во ориентацијата (Risojevic et al. 2012) и подобрен Габор дескриптор на текстурата (Enhanced Gabor Texture Descriptor (EGTD)) (Risojevic et al. 2013). Chen C. et al. (2015) предлагаат комплетни локални бинарни примероци во повеќе размери за класификација на слики снимени од далечина и постигнале одлични резултати споредено со другите методи на ниско ниво.

Методите кои користат визуелна репрезентација на средно ниво се друга група на методи за претставување на сцени, кои се обидуваат да ги прикажат сцените со статистичка репрезентација од висок степен на локално екстрахираните карактеристики на сликите. Тие прво изведуваат екстракција на локалните карактеристики на сликата од локалните закрпи (парчиња), со употреба на дескриптори како хистограми на боја или трансформација на карактеристики независно од размерот. Потоа, овие карактеристики се кодираат за да се добие репрезентација на средно ниво за сликите снимени од далечина. Вообичаено користен метод на средно ниво е вреќа на визуелни зборови (bag-of-visual-words (BoVW)) (Yang et al. 2010). Овој метод прво ги опишува локалните делови на сцената со дескриптори добиени со трансформација на карактеристики независно од размерот (Lowe 2004) и потоа учи речник на визуелни зборови, на пример користејќи кластерирање по најблиска средна вредност (k-means clustering). Речникот се нарекува визуелен речник или визуелна кодна книга. Оваа техника на средно ниво и нејзините варијации често се погодни за претставување на сцените снимени од воздух. Кај задачите за класификација на воздушни сцени, основниот метод вреќа на визуелни зборови може да се користи со различни локални дескриптори (Chen et al. 2011): трансформација на карактеристики независно од размерот, локални бинарни примероци, хистограми на боја, GIST. За класификација на сцени може да се

користат повеќе сликови карактеристики, со помош на хиерархиски методи на класификација и нормализација со теорија на екстремни вредности (Extreme Value Theory (EVT), (Scheirer et al. 2012)). Друг пристап е со примена на метод на ретко кодирање (Sheng et al. 2012) каде што структурните, спектралните и текстурните карактеристики се екстрахираат и се кодираат. Одлични резултати може да се постигнат со намалување на димензионалноста со анализа на главните компоненти (Principal Component Analysis (PCA)), пред да се спојат различните карактеристики, или со методи како подобрен Fisher вектор (Improved Fisher Vector (IFV)) (Perronnin et al. 2010) и вектори на локално агрегирани тенсори (Vectors of Locally Aggregated Tensors (VLAT)) (Negrel et al. 2014). Методите кои се споменати погоре не ја вклучуваат просторната распределба на визуелните зборови, која според некои автори е важна. Имено, Yang et al. (2011) го предлагаат методот јадро на ко-појава на просторната пирамида (Spatial Pyramid Co-occurrence Kernel (SPCK)), кој наспроти основниот модел на методот вреќа на визуелни зборови, ги интегрира апсолутните и релативните просторни податоци. Овој метод ги вклучува принципите на два други метода: јадро на совпаѓање на просторната пирамида (Spatial Pyramid Match kernel (SPM)) (Lazebnik et al. 2006) и јадро на просторна ко-појава (Spatial Co-occurrence Kernel (SCK)) (Yang et al. 2010). Chen S. et al. (2015) развиле модел кој ги интегрира апсолутните и релативните просторни врски од локалните прикази на ниско ниво во моделот пирамида на просторни релации (Pyramid-of-Spatial-Relations (PSR)). Покрај методот вреќа на визуелни зборови, постојат и класификациски модели на средно ниво кои ја земаат предвид семантичката врска меѓу локалните визуелни зборови на ниско ниво за претставување на сликата. CIE Lab моментите на боја (Stricker et al. 1995) се користат со латентна Дирихлетова алокација (Latent Dirichlet Allocation (LDA)) (Chen C. et al. 2015; Blei et al. 2003). Матрицата на истовремена појава на нивото на сиво (Haralick et al. 1973) и хистограмот на ориентација на рабовите (Edge Orientation Histogram (EOH)) (Kusumaningrum et al. 2014) се други начини за екстракција на различни информации кај моделот на латентна Дирихлетова алокација. Техниката веројатносна латентна семантичка анализа (probabilistic Latent Semantic Analysis (pLSA)) (Bosch et al. 2006) е усвоена од Zhong et al. 2015.

2.3 Класификација на воздушни сцени со методи на „длабоко“ учење

Третата група на техники за класификација на воздушни сцени се базира на визуелни информации на високо ниво. Споредено со методите кои се изложени претходно во оваа секција, методите на високо ниво може да екстрахираат поапстрактни и дискриминаторски семантички претстави. Најновите техники кои вклучуваат методи на „длабоко“ учење базирани на конволуциски невронски мрежи (Convolutional Neural Network (CNN)) покажуваат значително подобрување на точноста на класификација споредено со постарите методи базирани на рачно екстрахирани карактеристики (Cheng et al. 2020; Khelifi et al. 2020). Постојат многу задачи од компјутерски вид кои можат успешно да се решат со користење на методите на „длабоко“ учење: класификација на слики, препознавање на објекти и пребарување на слики. Постојат бројни научни студии кои покажуваат дека екстракцијата на карактеристики со конволуциски мрежи тренирани на масивни податочни множества (како ImageNet, Russakovsky et al. 2015) може успешно да се искористат за извлекување на карактеристики од слики во различни задачи (Razavian et al. 2014). Одлични експериментални резултати се постиганти при класификација на сцени снимени од воздух (Penatti et al. 2015; Hu et al. 2015; Nogueira et al. 2017).

Првиот случај на усвојување на претходно тренирана конволуциска мрежна архитектура за класификација на слики снимени од далечина е изведено од Penatti et al. 2015, каде што претходно тренираните AlexNet и Overfeat (Sermanet et al. 2013) се користат за екстракција на карактеристики. Овде активациите од првото целосно поврзано ниво од архитектурата на конволуциската мрежа се употребени за презентација на сликите. Penatti et al. (2015) пријавиле одлични резултати за две податочни множества од воздушни сцени, надминувајќи неколку рачно креирани визуелни дескриптори. Најпопуларен пристап за екстракција на карактеристики е да се вклучат карактеристики од погорните конволуциски нивоа, или од последното целосно поврзано ниво кое му претходи на нивото за класификација. Меѓутоа, кога целната задача која е од интерес значително се разликува од оригиналната задача, карактеристиките кои се извлекуваат од подолните конволуциски нивоа се покажале како несоодветни (Yosinski et al. 2014). Hu et al. (2015) со помош на конволуциски невронски мрежи екстрахирале карактеристики во повеќе размери од последното конволуциско ниво, а потоа ги кодирале со техниката вреќа на

визуелни зборови (Sivic et al. 2003), вектор на локално агрегирани дескриптори (Vector of Locally Aggregated Descriptors (VLAD)) (J'egou et al. 2012) и подобро Fisher јадро (Perronnin et al. 2010) за да се формира конечен приказ за сцената. Уште користеле и карактеристики од две претходно тренирани мрежи и на тој начин извеле класификација на воздушни сцени со висока резолуција. Тие предлагаат карактеристики кои се добиени со фузија на активациите од средните нивоа и од последното целосно поврзано ниво од мрежните шеми. Пред да се изведе фузијата, на активациите од конволуциските нивоа се применуваат алгоритми за кодирање на карактеристиките. Глобални карактеристики се екстрахирани од Nogueira et al. (2017) со помош на конволуциските мрежни модели и тие се спроведени на класификатор. Во сите овие експерименти, глобално или локално екстрахираните карактеристики се добиени од невронски мрежи претходно тренирани на податочни множества со природни слики. Овие карактеристики се користат за класификација на слики снимени од далечина.

Постојат многу слободно достапни претходно тренирани длабоки конволуциски архитектури: ResNet, DenseNet, Inception, Xception, итн. Најшироко користени мрежни модели за класификација на сцени снимени од воздух се CaffeNet, GoogleNet и VGGNet (Nogueira et al. 2017; Cheng et al. 2018; Chaib et al. 2017; Wang et al. 2018; Yu et al. 2018a). Овие невронски мрежи се состојат од 30, па и повеќе нивоа и имаат огромен број на параметри. Точноста на класификација на сликите снимени од далечина може да се зголеми со стратегија за влез на повеќе нивоа за учење на конволуциски мрежи со „повеќе погледи“, слично како што Luus et al. (2015) покажале за GoogLeNet. Chatfield et al. (2014) ги проценувале длабоките карактеристики за класификација на традиционални слики, самостојно или во комбинација со други карактеристики. VGGNet се користи за извлекување на карактеристики од различни мрежни нивоа, а потоа истите се трансформираат со анализа на дискриминантна корелација (Discriminant Correlation Analysis (DCA)) (Chaib et al. 2017). Трансформираните карактеристики се спојуваат и потоа се применува машина со вектори за поддршка (Support Vector Machine (SVM)) за класификација (Chen Y. et al. 2018). Образложението на овој процес е да се користи конволуцијата како ефективен начин за екстракција на нова, компактна и ефективна претстава на карактеристиките од необработените податоци, поедноставувајќи ја задачата

на класификација која следи. Оваа способност на невронските мрежи успешно се користи за екстрахирање на вектори на карактеристики во проблеми на предикција во контекст на податоци на граф (Li et al. 2018, Pio et al. 2020) и податоци од временски серии (Corizzo et al. 2019; Bao et al. 2017). Фузија на карактеристики може да се најде и во други трудови (Yu et al. 2018a; Corizzo et al. 2020).

Yu and Liu (2018a) предлагаат две шеми. Првата ја користи оригиналната мрежа за екстракција на карактеристики од RGB сликите, додека пресликаната кодирана мрежа на локални бинарни примероци (Local Binary Pattern (LBP)) се користи за екстракција на карактеристики од мапите на карактеристики со локални бинарни примероци. По овој чекор, фузија на карактеристиките се изведува од слојот за спојување: карактеристиките минуваат низ целосно поврзаните нивоа и на крајот се класифицираат. Втората шема користи мрежа кодирана по значајност (saliency) наместо пресликана кодирана мрежа на локални бинарни примероци. Wang Q. et al. (2018) користат рекурентни невронски мрежи (Recurrent Neural Networks (RNNs)) за класификација на слики снимени од далечина. Рекурентните невронски мрежи се вклучени во изградба на механизмот на внимание. Cheng et al. (2018) презентирале нова критериумска функција, со примена на метричко учење на карактеристиките од конволуциските мрежи. Метричката критериумска функција е комбинирана со стандардната функција за оптимизација (крос-ентропијата). Овој пристап резултира во тоа карактеристиките кои припаѓаат на слики од истата класа да бидат многу слични, додека карактеристиките на сликите од различни класи да се разликуваат.

Точноста на класификација на претходно тренираните конволуциски мрежи може понатаму да се подобрува преку фино подесување на тежините. Финото подесување е метод на пренос на учење кој ги прилагодува параметрите на претходно тренираната мрежа со дополнително тренирање со новото податочно множество, при што се адресира нова задача со различен број на класи од иницијалното излезно ниво на оригиналната мрежна архитектура. Финото подесување ја тренира мрежата со мала почетна стапка на учење и редуциран број на епохи за тренирање, споредено со комплетен процес на тренирање од почеток. Во тек на овој процес, критериумската функција постигнува подобар минимум споредено со тренирање со случајна

иницијализација на тежините. Girshick et al. (2014) покажале дека значително подобрување на перформансите може да се добие со фино подесување на претходно тренираната конволуциска мрежа. Тие фино ја подесувале AlexNet (Krizhevsky et al. 2012) и постигнале подобри резултати за семантичка сегментација. Неколку други трудови (Yue et al. 2015; Xie et al. 2015) во научната заедница исто така ги проучуваат предностите на финото подесување на конволуциските невронски мрежи. Xie et al. (2015) проценувале целосно тренирана конволуциска мрежа споредено со фино подесена, за да се откријат недостатоците во контекст на воздушни сцени. Пристапот презентирани од Yue et al. (2015) ангажира техника на фино подесување за класификација на хиперспектрални слики. Castelluccio et al. (2015) предлагаат фино подесување на тежините на конволуциските слоеви од претходно тренираните мрежи за извлекување на подобри сликови карактеристики. Експерименталните резултати прикажани кај Hu et al. (2015) и Nogueira et al. (2017) покажуваат дека финото подесување на конволуциски мрежи претходно тренирани на ImageNet дава добра точност на класификација на податочни множества од воздушни сцени.

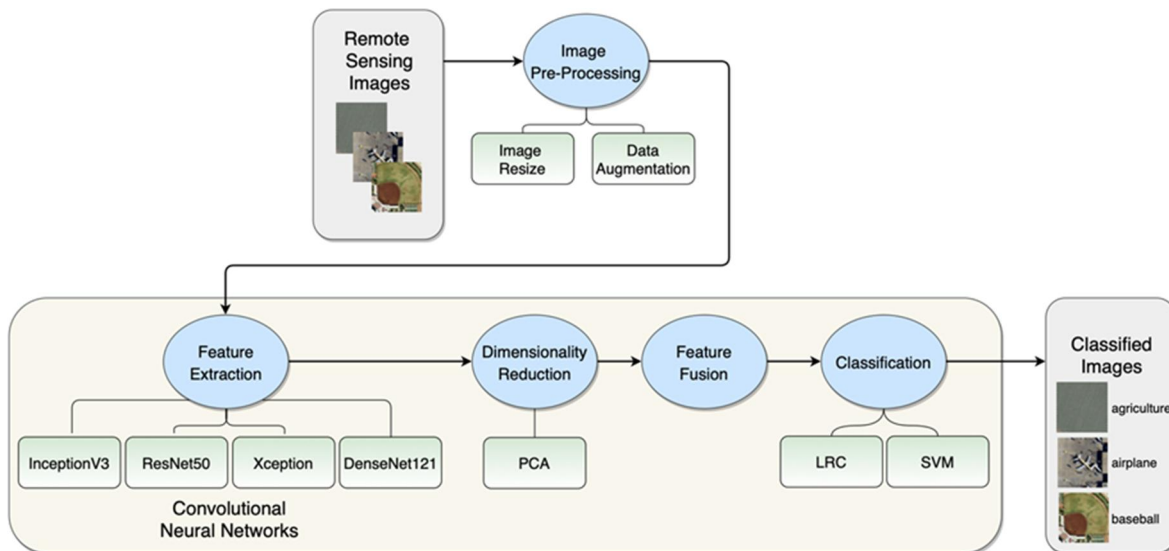
Со цел да се проценат различните техники кои користат длабоки невронски мрежи, Jarrett et al. (2009) ги евалуирале најдобрите шеми и методи за тренирање, за мрежи тренирани со и без учител. Larochelle et al. (2009) се обиделе да го одредат оптималниот начин за тренирање на невронски мрежи, вклучувајќи и „алчно“ тренирање слој-по-слој (greedy layer-wise) и тренирање без учител. Спротивно на методите за пренос на учење презентирани во оваа секција, алтернатива е да се оптимизира конволуциската мрежа од почеток, односно со случајна распределба на параметрите (тежините). Меѓутоа, според Nogueira et al. (2017), користењето на UC Merced податочното множество (Yang et al. 2010) и WHU RS податочното множество (Xia et al. 2010) за целосно тренирање на конволуциски невронски мрежи како CaffeNet (Jia et al. 2014) или GoogLeNet (Szegedy et al. 2014) покажува лоши резултати при класификација. Причината за лошите резултати и слабата генерализација лежи во тоа што големите конволуциски архитектури имаат голем број на параметри и тренирањето на моделот со податочни множества на сцени снимени од воздух кои бројат неколку илјади слики неизбежно заглавува во локален минимум. Додека, тренирањето на ваквите мрежи од почеток со употреба на поголеми

податочни множества како AID (Xia et al. 2017) и NWPU-RESISC45 (Cheng et al. 2017a) може да покаже добри резултати.

3. ЦЕЛ НА ИСТРАЖУВАЊЕТО

Во докторската теза се користат архитектури за класификација на воздушни сцени кои ја користат моќта на претходно тренирани конволуциски мрежи кои се способни да класифицираат воздушни сцени со висока резолуција. Експерименталната поставка е поделена во два дела: методи кои користат екстракција на карактеристики и методи кои користат фино подесување.

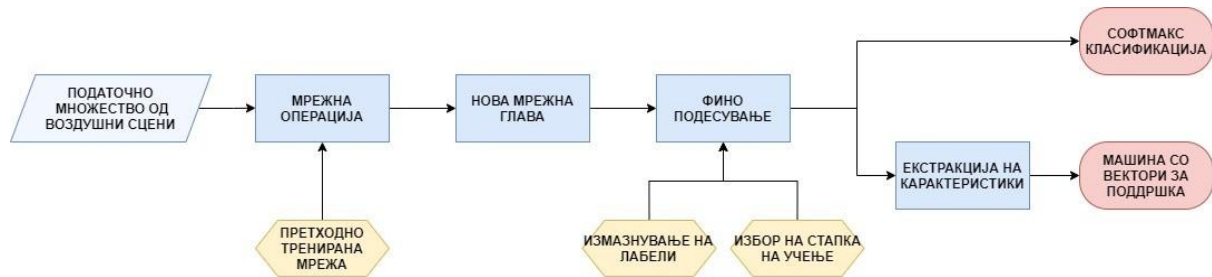
Нашиот пристап со екстракција на карактеристики се обидува да создаде подобра претстава за сцените снимени од воздух од активациите на невронски мрежи. Конволуциски невронски мрежи кои се вклучени во експериментите се: ResNet50 (He et al. 2015), InceptionV3 (Szegedy et al. 2015), Xception (Chollet 2017), и DenseNet121 (Huang et al. 2018). Моделите се целосно тренирани на ImageNet податочното множество (Russakovsky et al. 2015). Овие конволуциски мрежи изведуваат екстракција на карактеристики со отстранување на одредени нивоа на оригиналната претходно тренирана мрежа. Претходно тренираните мрежи имаат комплексна архитектура со десетици слоеви и екстракцијата на карактеристики се прави од различни слоеви, како кај Hu et al. (2015). Овде се користат активациите од нивото за редукција на просек (average pooling layer), последното конволуциско ниво и некои од средните конволуциски нивоа, со цел да се добијат репрезентативни карактеристики за воздушните сцени. Со оваа постапка од конволуциските нивоа се добиваат вектори на карактеристики со значителни димензии. Од тие причини, се користат методи за редукција на димензиите пред овие карактеристики да се спојат со карактеристиките екстрахирани од нивоата за редукција на просек. По екстракцијата на карактеристики и нивната фузија, има потреба од класификатор за да се добијат семантичките лабелите на сликите снимени од далечина. Овде предлагаме два широко користени линеарни класификатори – линеарна машина со вектори за поддршка и класификатор со логистичка регресија (Logistic Regression Classifier (LRC)) - за да се процесираат екстрахираните карактеристики и да се класифицираат сцените. Шематски приказ на предложениот метод со екстракција на карактеристики е прикажан на слика 3.1.



Слика 3.1 Дијаграм на предложениот метод со екстракција на карактеристики
 Figure 3.1 Diagram of the proposed feature extraction method

Во докторскиот труд за разрешување на проблемот на класификација на воздушни сцени со висока резолуција со методот на fino подесување користени се четири различни мрежни архитектури. ResNet50, InceptionV3, Xception и DenseNet121 се усвоени со цел да се одреди нивната ефективност во задачите на класификација на слики снимени од далечина. Прво, испитувано е финото подесување на тежините согласно конкретното податочно множество од слики снимени од воздух. Во процесот на fino подесување, отстранети се последните слоеви на секоја од претходно тренираните мрежи по нивото за редукција на просек (т.н. „мрежна операција“) и додадена е нова глава на мрежата. Новата мрежна глава се состои од: целосно поврзано ниво, ниво за испуштање и софтмакс (softmax) ниво. Тренирањето се изведува на модифицираната длабока невронска мрежа. Потоа, fino подесените конволуциски мрежи се експлоатираат за екстракција на карактеристики и екстрахираните карактеристики се употребуваат за тренирање на машини со вектори за поддршка (Support Vector Machine (SVM)), кои досега се успешно применувани во други проблеми за класификација на слики и пренос на учење (Corizzo et al. 2019, Mignone et al. 2019). Во овој случај машината е имплементирана во две верзии: со линеарно јадро и јадро со радијални основни функции (Radial Basis Function (RBF)). Користиме стапка на учење со линеарно опаѓање и циклични стапки на учење и ја евалуираме нивната погодност за fino

подесување на претходно тренирани конволуциски мрежи за класификација на воздушни сцени. Уште повеќе, применето е и измазнување на лабели (label smoothing) (Müller et al. 2015), како регуларизациона техника и се проценува нејзиното влијание на точноста на класификација споредена со современите методи. Сликата 3.2 го прикажува дијаграмот на предложениот метод.



Слика 3.2 Дијаграм на предложениот метод со фино подесување

Figure 3.2 Diagram of the proposed fine-tuning method

Главните цели на оваа докторска теза се рекапитулирани како што следи:

- да се споредат претходно тренираните конволуциски мрежи ResNet50, InceptionV3, Xception и DenseNet121 при примена на методите за пренос на учење за класификација на воздушни сцени: при извлекување на сликови карактеристики, како и при фино подесување на невронските мрежи;
- да се анализира методот на екстракцијата на карактеристики со ангажирање на редукционите нивоа на просек и конволуциските нивоа (последното или некое од средните) на секој од предложените мрежни модели;
- да се евалуира фузијата на карактеристики добиени од активациите на конволуциските и редукционите нивоа, базирано на перформансите на лиенарните класификатори: линеарна машина со вектори за поддршка и класификатор со логистичка регресија. Презентираната техника за екстракција на карактеристики со употреба на претходно тренирани невронски мрежи вклучува и редукција на димензиите на густите мрежни активации од конволуциските нивоа (последното или едно од средните конволуциски нивоа) со употреба на анализа на главните компоненти;
- предложената техника со екстракција на карактеристики да се спореди со постоечките методи на две јавно достапни податочни множества од

воздушни сцени: UC Merced податочното множество и WHU RS податочното множество;

- да се анализира влијанието на адаптивните стапки на учење: стапката со линеарно опаѓање и цикличните стапки на учење кај методот со фино подесување од аспект на точноста на класификација;
- да се процени влијанието на измазнувањето на лабели на способноста на моделот за генерализација споредено со современите техники за класификација на воздушни сцени;
- кај методот со фино подесување на тежините на мрежата да се евалуира преносливоста на карактеристиките добиени од фино подесените конволуциски мрежи и нивна класификација со линеарни и машини со вектори за поддршка со јадро со радијални основни функции;
- евалуација на методот со фино подесување да се изврши на множествата на слики: AID и NWPU-RESISC45 и експерименталните резултати да се споредат со постигнатите точности на класификација од други автори во сродни написи.

Со предложените методи на пренос на учење (transfer learning) се очекува да се добијат точности на класификација на различните податочни множества од слики снимени од далечина кои се споредливи со постоечките методи на класификација, со можност да се обезбеди нова референца за класификација на множествата од воздушни сцени. При одредени симулациони сценарија дури се очекува и да се надминат постоечките методи, или барем поголем дел од нив. Поважно би било да се одредат тенденциите при класификација со одредени претходно тренирани конволуциски мрежи, како и да се утврди под кои услови точноста на класификација би можела понатаму да се подобрува.

4. МЕТОДИ НА ИСТРАЖУВАЧКАТА РАБОТА

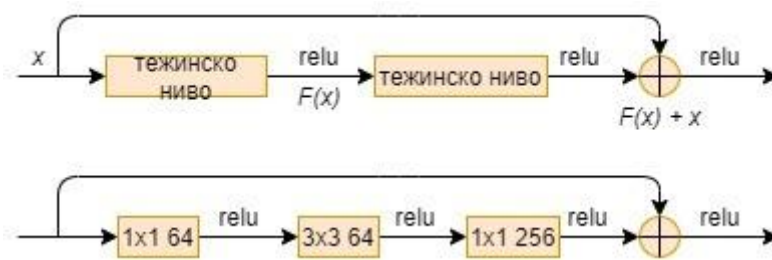
Конволуциските невронски мрежи се погодни за решавање на многу проблеми поврзани со слики, како сегментација на слики, класификација и детекција на објекти. Моделите на мрежите се структури изградени од различни нивоа, споени едно на друго. Слоевите се состојат од неврони кои можат да учат низ различните алгоритми за оптимизација. Овде во експериментите се користат четири различни архитектури: ResNet50, InceptionV3, Xception и DenseNet121.

Во оваа секција опишани се сите претходно тренирани конволуциски мрежи кои се користат за трансфер на учење: со екстракција на карактеристики и со фино подесување. Овие модели на архитектури ги користиме за пренос на учење бидејќи тие покажале врвни резултати при класификација на слики од ImageNet податочното множество. Потоа, воведени се класификатори кои се користат за класификација на сликите снимени од далечина: класификатор со логистичка регресија (Logistic Regression Classifier (LRC)) и машина со вектори за поддршка (Support Vector Machine (SVM)), како и анализа на главните компоненти (Principal Component Analysis (PCA)) која се користи за редукција на димензионалноста. Овде се разгледувани слични техники за моделирање на екстракција на карактеристики како и кај Yu et al. (2018b). Претставени се и адаптивните стапки на учење: со линеарно опаѓање и циклични стапки на учење, како и регуларизационата техника измазнување на лабели (label smoothing). Понатаму, опишани се четирите јавно достапни податочни множества, кои се користени во нашето истражување: UC Merced, WHU RS, AID и NWPU-RESISC45. Овие податочни множества се користат со цел да се евалуира нашиот пристап и да се спореди со слични написи. На крајот од оваа секција се дадени описите на експерименталните поставки и метриката која се користи за евалуација.

4.1 ResNet

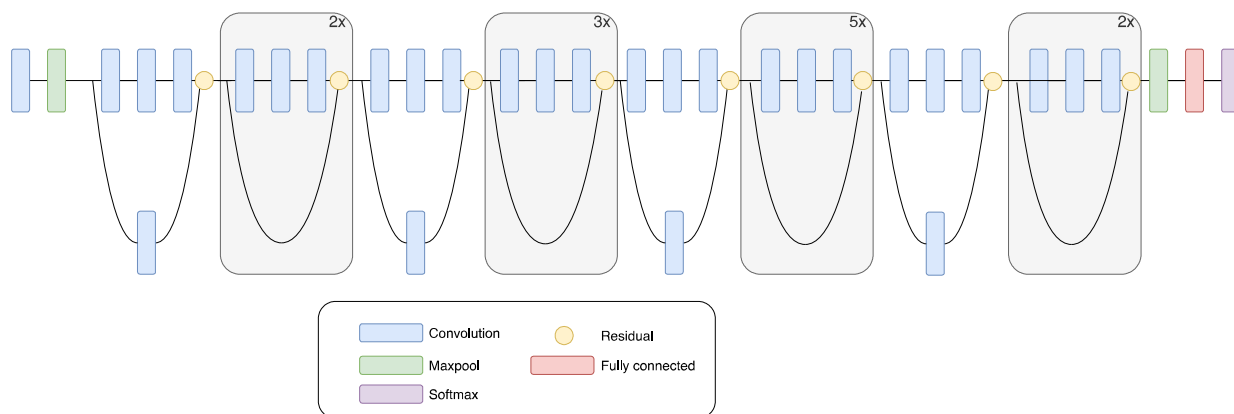
ResNet го освоила натпреварот од големи размери за визуелно препознавање на ImageNet во 2015 година (ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC-2015)) во делот на задачата за класификација. ResNet е многу длабока конволуциска мрежа која може да има до 152 слоја (He et al. 2015). Два главни проблема се поврзани со тешкотиите во тренирањето на длабоки архитектури: големата грешка при тренирање и

проблемот со исчезнување на градиентот. Проблемот со исчезнувањето на градиентот предизвикува неефективност при тренирањето на мрежата. Оваа неефективност е присутна кај пониските слоеви за време на тренирањето. ResNet го решава проблемот на исчезнувањето на градиентот со примена на резидуален модул (residual module). Овој модул за „длабоко“ учење има кратка врска помеѓу влезот и излезот. Неговата намена не е да учи нелинеарна функција, туку остаток на функција, имено остатокот $F(x)$ меѓу излезот $F(x) + x$ и влезот x на модулот, како што е прикажано на сл. 4.1. Постојат две варијанти на резидуалниот модул: основна верзија и верзија „тесно грло“ (bottleneck). Основниот модул се состои од две 3×3 конволуциски нивоа. Верзијата „тесно грло“ на резидуалниот модул за учење дополнително содржи две 1×1 конволуциски нивоа и нивната цел е да ја намалат димензионалноста на податоците. Редукцијата на димензионалноста води кон намален број на мрежни тежини, со што се намалува комплексноста на пресметките за време на тренирање на мрежата, а со тоа се овозможуваат многу длабоки архитектури, како ResNet152 (He et al. 2015).



Слика 4.1 Резидуален модул (горе) и модул „тесно грло“ (долу) кај ResNet
 Figure 4.1 Residual block (top) and “bottleneck” block (bottom) of ResNet

Резидуалниот модул има свое пресликување, кон кое се прилагодуваат сите слоеви во модулот. Тренирањето на мрежата е поедноставно на мапата на модулот, споредено со изворната мрежна структура. ResNet архитектурата е слична со VGG моделот поради тоа што содржи претежно 3×3 конволуциски филтри. Меѓутоа, ResNet има помалку филтри споредено со VGG мрежата и затоа е поедноставна. На сл. 4.2 е прикажан шематскиот дијаграм на ResNet.



Слика 4.2 Шематски приказ на ResNet конволуциската мрежа
 Figure 4.2 Schematic drawing of ResNet CNN

4.2 Inception

Интуицијата позади овој вид на мрежи се заснова на фактот дека корелацијата меѓу мрежните пиксели е локална. Земајќи ги предвид локалните корелации меѓу мрежните пиксели, можно е да се намали бројот на параметри за учење. Првата длабока конволуциска мрежа од Inception множеството е наречена InceptionV1 (Szegedy et al. 2014) и е претставена како GoogLeNet. GoogLeNet е конволуциска невронска мрежа која го освоила натпреварот за визулено препознавање на ImageNet во 2014 година во делот за класификација и детекција на објекти. GoogLeNet е несеквенцијална конволуциска мрежа. GoogLeNet го решава прашањето на намалување на бројот на параметри за учење со вклучување на „воведен“ (inception) модул во дизајнот на мрежната архитектура, како што е прикажано на сл. 4.3. Тој произлегува од идејата дека повеќекратните врски меѓу слоевите водат кон двојност (редундантност), поради корелацијата меѓу нив. И самиот „воведен“ модул е конволуциска мрежа. Тој се состои од 22 нивоа и го процесира својот влез на паралелен начин. Во средните слоеви се вклучени неколку помошни класификатори. Помошните класификатори се вметнати за да се зајакне способноста за дискриминација на подолните слоеви. Овој модул може да користи операции на конволуција и редукција во секое ниво. На пример, кај AlexNet и VGG секое ниво користи или операција на конволуција или операција на редукција. Главна карактеристика на „воведниот“ модул е дека во рамки на ист слој се користат филтри со различни димензии: 1×1 , 3×3 и 5×5 . Ова води кон различни димензии на екстрахираните примероци, како и поисцрпни информации. Слојот на „тесно грло“ (bottleneck

layer), кој е 1×1 конволуциски слој, има двојна функција: да ги поедностави пресметките и да го намали бројот на параметри на конволуциската мрежа. Конволуциските нивоа 1×1 им претходат на 3×3 и 5×5 конволуциските нивоа. По 1×1 конволуциските нивоа се користи ReLU и целта на оваа операција е да се зголеми нелинеарноста и да се продлабочи мрежата. Филтрите со различни димензии се користат да се покрие поголемо рецептивно поле од секој кластер. Излезите од различните нивоа потоа се спојуваат и тоа претставува излез од модулот.

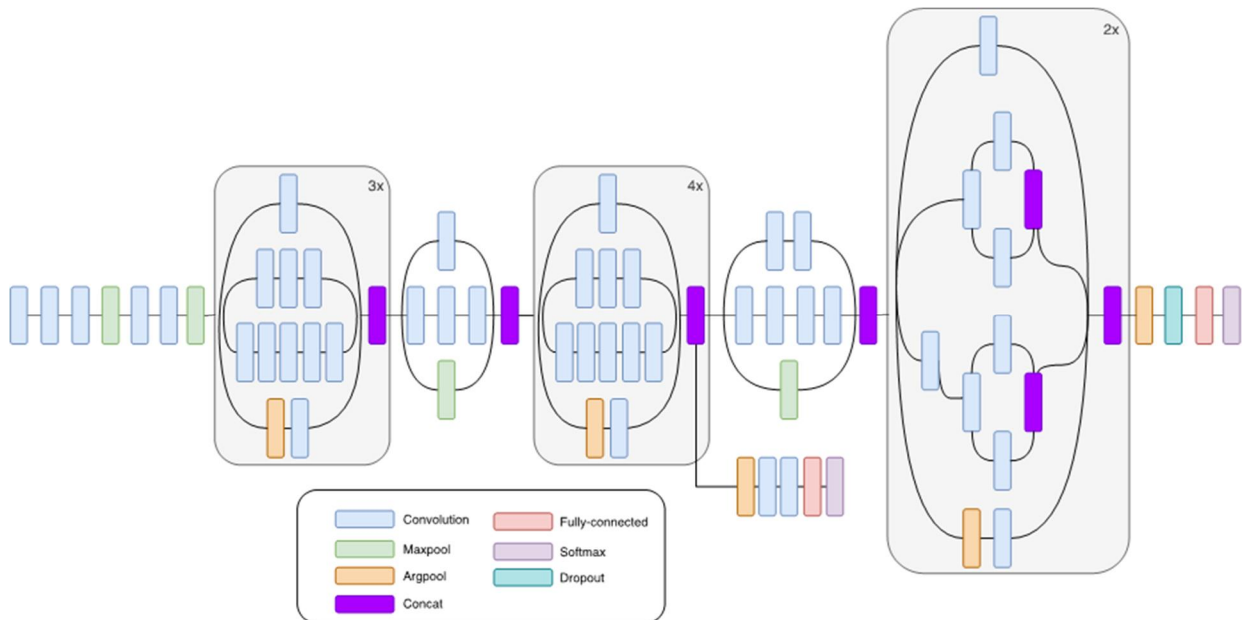


Слика 4.3 Архитектура на основниот „введен“ модул

Figure 4.3 Architecture of the basic inception module

Воведувањето на нормализација на групи (batch normalization) во мрежната архитектура (Szegedy et al. 2014; Ioffe et al. 2015) резултирало во InceptionV2 модел. Третата итерација, која се нарекува InceptionV3 (Szegedy et al. 2015), добиена е со дополнителни процедури на факторизација. Овој процес резултира во три различни „введени“ модули: „введен“ модул тип 1, добиен со факторизација во помали конволуции; „введен“ модул тип 2, кој се постигнува со факторизација во асиметрични конволуции и „введен“ модул тип 3, кој е воведен за да се подобрат презентациите со големи димензии. Кај оваа мрежа нема целосно поврзани нивоа, наместо нив се користи ниво за редукција на просек. Отсуството на целосно поврзани нивоа го намалува бројот на параметри. InceptionV3 мрежата има помалку параметри споредено со AlexNet и

VGG, а сепак има способност да учи подлабоки презентации на карактеристиките (Chollet 2017). Сл. 4.4 го прикажува дијаграмот на InceptionV3 конволуциската мрежа.



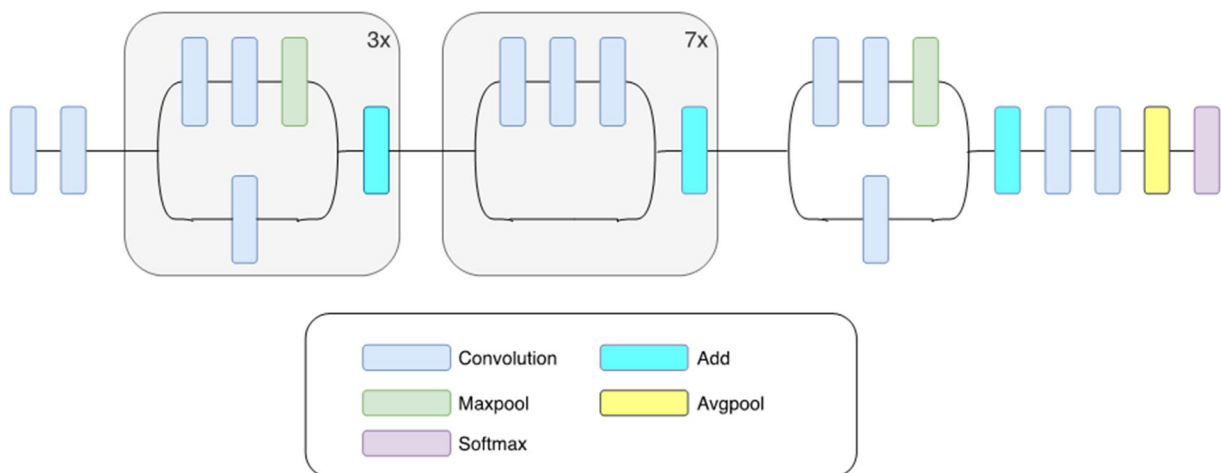
Слика 4.4 Шематски приказ на InceptionV3 конволуциската мрежа

Figure 4.4 Schematic drawing of InceptionV3 CNN

4.3 Xception

Друга длабока мрежа која е слична на GoogLeNet е Xception. Кај Xception, наместо „воведниот“ модул се користат конволуциски нивоа разделливи по длабочина (depthwise separable) (Chollet 2017). Тој (Chollet 2017) претпоставува дека е добра операција да се раздвојат пресликувањето на меѓуканалните корелации и просторните корелации во мапите на карактеристики од мрежната архитектура. Оваа теза е посилен верзија од тезата на која се темелат мрежите од Inception множеството. Од оваа причина, Chollet (2017) ја нарекол оваа конволуциска архитектура Xception (Extreme Inception). Тој предложил подобрување на мрежите од Inception множеството со замена на „воведните“ модули со конволуции разделливи по длабочина. Оваа мрежа е композиција од разделливи по длабочина конволуциски нивоа со резидуални конекции (како што е прикажано на сл. 4.5). Конволуцијата разделлива по длабочина, која е исто така позната како „разделлива“ конволуција, се изведува во два чекора. Првиот чекор е конволуција по длабочина (depthwise) (Mahdianpari et al. 2018), или

просторна конволуција имплементирана на секој влезен канал посебно. Вториот чекор е конволуција по точки (pointwise). Тоа е 1×1 конволуција која го води излезот од каналите добиени со конволуцијата по длабочина кон нов простор на каналите.



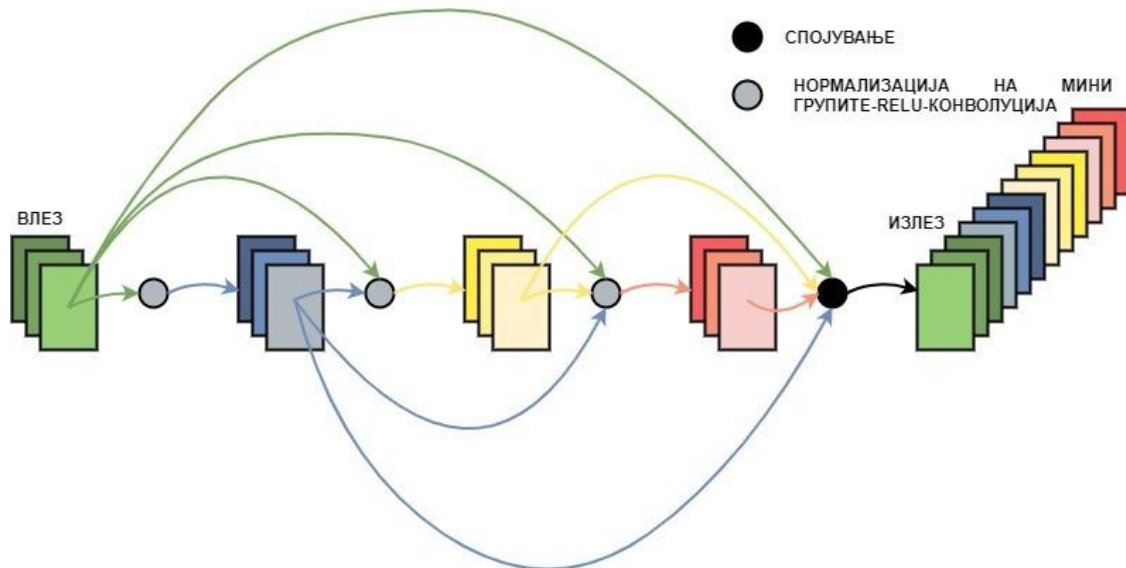
Слика 4.5 Шематски приказ на Xception конволуциската мрежа

Figure 4.5 Schematic drawing of Xception CNN

4.4 DenseNet

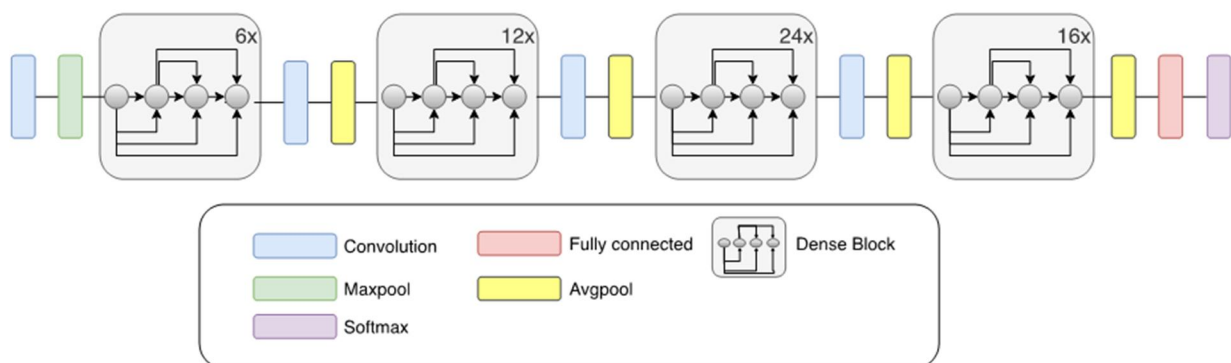
Исчезнување на градиентот е проблем поврзан со тренирање на мрежата кој произлегува од нејзината длабочина. DenseNet, исто како и ResNet, се бори со овој проблем. Архитектурата на DenseNet се базира на конекција на сите слоеви со соодветни димензии на мапите на карактеристики, директно еден со друг во манир на проследување нанапред (feedforward), што обезбедува најдобар проток на информации меѓу слоевите (Huang et al. 2018). Во мрежната структура на DenseNet, секој слој добива влезови од сите претходни слоеви и го поврзува својот излез со секое следно ниво. Мапите на карактеристики од секое ниво се сериски поврзани за да носат информации од претходните нивоа кон следните нивоа. Сл. 4.6 го покажува шематски образецот на поврзување. Стрелките со различна боја го прикажуваат влезот и излезот од одредено мрежно ниво. Како што може да се види на истата слика, сите мапи на карактеристики поминуваат низ операциите нормализација на групите – „исправена линеарна единица” (ReLU) - конволуција и на крај се спојуваат. Кога се споредува со ResNet, пристапот на Huang et al. (2018) не изведува собирање

на карактеристиките за потоа да се водат на следното ниво. Спротивно на тоа, тој ги комбинира карактеристиките со спојување (конкатенација). Како што може да се види од шематскиот приказ, образецот на спојување е густ и тоа резултира во името на мрежата.



Слика 4.6 Густо поврзан шаблон на конволуција
Figure 4.6. Densely concatenated convolution pattern

Оваа конволуциска мрежа содржи помалку параметри од другите мрежи, бидејќи употребата на густ шаблон на поврзување имплицира дека нема потреба повторно да се учат двојни (редундантни) мапи на карактеристики. DenseNet121, како специфична имплементација на DenseNet која се користи во овој труд, покажува одлични резултати при класификација со мали податочни множества за тренирање и не е склona кон претренирање (overfitting) (Huang et al. 2018). На сл. 4.7 е прикажан дијаграм на DenseNet.



Слика 4.7 Шематски приказ на DenseNet конволуциската мрежа
Figure 4.7 Schematic drawing of DenseNet CNN

4.5 Класификатор со логистичка регресија (Logistic Regression Classifier (LRC))

Логистичката регресија е метод на машинско учење со учител кој се користи во проблеми на класификација. Влезот X е матрица која содржи N податоци, секој претставен со K карактеристики. Влезовите x_{ij} се вектори на карактеристики со должина K (по x_i), кои се континуирани, со индекси $j = 1, 2, \dots, K$ и $i = 1, 2, \dots, N$. Излезот y_j има вредност во интервалот $\{0, 1\}$ и е бинарна варијабла, со Бернулиева распределба и параметар p_i . Функцијата на активација на линеарниот класификатор со логистичка регресија се нарекува „логистичка функција“ (сигмоидална функција). Главната карактеристика на сигмоидалната функција е да се ограничи излезот на еден систем во интервалот $\{0, 1\}$, без разлика на варијаблите на неговиот влез. Постериорните (posterior) веројатности изразени преку логистичката функција се дадени со:

$$P(Y/X) = \frac{1}{1+e^{-f(x)}}. \quad (4.1)$$

Во изразот (4.1) карактеристиките (x_j) и соодветните тежини (β_j) ја сочинуваат функцијата $f(x)$. Оптимизацијата на класификаторот со логистичка регресија се изведува со методот на максимална веродостојност (Maximum Likelihood Estimation (MLE)). Целта при оптимизација е да се направи веројатноста во процесот на тренирање што е можно поголема и истата е изразена со:

$$\operatorname{argmax}_{\beta}: \log\left\{\prod_{i=1}^n P(y_i/x_i)^{y_i}(1 - P(y_i/x_i))^{(1-y_i)}\right\}. \quad (4.2)$$

Нејзини параметри се: излезот y кој е во интервалот $\{0,1\}$, $P(y_i/x_i)$ е постериорната веројатност и е дадена со $\frac{1}{1 + e^{-f(x)}}$ и векторот на тежините кај $f(x) - \beta$.

4.6 Машина со вектори за поддршка (Support Vector Machine (SVM))

Машината со вектори за поддршка е дискриминативен класификатор кој е дефиниран со хипер-рамнина на раздвојување. За определено множество на податоци за тренирање, машината наоѓа оптимална хипер-рамнина која ги категоризира новите примероци. Хиперпараметрите кои може да се подесуваат кај овој модел се: видот на јадрото, влијанието на регуларизацијата, гама (gamma) и маргината. Јадрото може да биде линеарно, полиномно,

експоненцијално, Гаусово итн. За линеарно јадро, предикцијата за влезот се пресметува со:

$$f(x) = B(0) + \sum(a_i * (x, x_i)) \quad (4.3)$$

Таа се добива со скаларен производ на новиот влезен вектор (x) со секој од векторите за поддршка (x_i) кај податоците за тренирање. Коефициентите $B(0)$ и a_i (за секој влез) се проценуваат од податоците за тренирање од страна на алгоритмот за учење. Гаусовото јадро и експоненцијалното јадро се дадени со:

$$K(x, x_i) = \exp(-\text{gamma} * \sum(x - x_i)) \quad (4.4)$$

$$K(x, x_i) = \exp(-\text{gamma} * \sum(x - x_i)^2) \quad (4.5)$$

Гаусовото јадро е пример за јадро со радијални основни функции. Во овој труд, при примена на методот за пренос на учење со екстракција на карактеристики ќе се користи машина со вектори за поддршка со линеарно јадро, а при примена на методот со фино подесување ќе се користи машина со вектори за поддршка со линеарно и со Гаусово јадро.

Регуларизациониот параметар дава информација за тоа до кој степен да се поништи погрешната класификација на секој примерок за тренирање при оптимизација на машината. Оптимизацијата со голем регуларизационен параметар цели кон подобра точност на класификација со користење на хипер-рамнина со помала маргина. Мала вредност за регуларизациониот параметар води кон хипер-рамнина за раздвојување со поголема маргина, што вообичаено резултира во намалена точност на класификација. Степенот на влијание на секој примерок за тренирање е одреден со гама параметарот. Малите вредности за гама имплицираат „далеку“, а големите - „блиску“.

4.7 Намалување на димензионалноста

Димензиите на екстрахираните карактеристики од нивоата за редукција на просек се еден или два реда на големини помали споредени со карактеристиките екстрахирани од конволуциските нивоа. Во овој труд, при примена на методот со екстракција на карактеристики, целта е да се овозможи еднакво влијание на двата типа на екстрахирани карактеристики на процесот на класификација: на оние од нивоата за редукција на просек, како и на оние од конволуциските нивоа. Дополнително, не сакаме карактеристиките од конволуциските нивоа да имаат

предоминантен ефект на точноста на класификација поради разликата во димензиите на векторите. Ова е причината поради која се изведува намалување на димензионалноста на карактеристиките од конволуциските нивоа со анализа на главните компоненти (Principal Component Analysis (PCA)), пред тие да се спојат со карактеристиките од нивоата за редуција на просек. Анализа на главните компоненти е техника за намалување на димензионалноста која одредува нова група на димензии на податоците (група на основи на гледање), која се користи за проекција на податоците од оригиналниот високодимензионален простор во приказ со помали димензии, притоа зачувувајќи го најголемиот дел од варијансата на податоците. Сите нови димензии се независни поради нивната ортогоналност и се рангираат во зависност од варијансата на податоците во нив. Првата главна компонента задржува најголема варијанса. Техниката е резимирана во следните чекори:

1. се пресметува матрицата на коваријанса X од влезните податоци со димензии $m \times n$;
2. следно треба да се пресметаат сопствените вектори (eigenvectors) и соодветните сопствени вредности (eigenvalues);
3. се редат сопствените вектори според нивните карактеристики, така што тие се намалуваат;
4. новите редуцирани k димензии ќе бидат првите k сопствени вектори;
5. Се трансформираат оригиналните n –димензионални податоци во k димензии.

Новата матрица има n податоци, секој од нив со по k карактеристики:

$$[\text{нови податоци}]_{k \times n} = [\text{првите } k \text{ сопствени вектори}]_{k \times m} [\text{оригинални податоци}]_{m \times n}. \quad (4.6)$$

Целта на анализата на главните компоненти е да се рашират податоците за да имаат голема варијанса и покрај малиот број на димензии и не треба да има коваријанса меѓу димензиите. Затоа, матрицата на коваријанса на трансформираниите податоци треба да биде дијагонална.

4.8 Линеарно опаѓање на стапката на учење

Најважните хиперпараметри при тренирање на конволуциската невронска мрежа се почетната стапка на учење, бројот на епохи за тренирање, распоредот

на стапката за учење, регуларизациониот метод (L2, dropout). Најголем дел од невронските мрежи се тренираат со алгоритмот стохастички градиентен спуст (Stochastic Gradient Descent (SGD)), кој ги ажурира мрежните тежини w со:

$$w += \eta \cdot \text{gradient}, \quad (4.7)$$

каде што η е стапка на учење, кој параметар ја одредува големината на чекорот на градиентот. Одржување на константна стапка на учење за време на тренирање на мрежата може да биде добар избор во некои ситуации, но многу често намалувањето на стапката за учење со тек на времето е поповолно. Кога се тренираат конволуциски невронски мрежи, ние се обидуваме да најдеме локален минимум, или само област каде што критериумската функција (функцијата на загуба) има доволно ниски вредности. Ако имаме константна но голема стапка на учење, не би било можно да се постигнат саканите вредности на функцијата на загуба. Во спротивно, ако ја намалиме стапката на учење, нашите конволуциски мрежи ќе бидат способни да се спуштат во пооптимални области на функцијата на загуба (Bengio et al. 2012). Во дел од нашите експерименти при примена на методот на фино подесување користиме распоред за линеарно опаѓање на стапката на учење, кој ја намалува нашата стапка на учење до нула на крајот на последната епоха на тренирање, како што е прикажано на сл.4.8. Стапката на учење η во секоја епоха на тренирањето е:

$$\eta = \eta_1 \cdot \left(1 - \frac{T}{T_{max}}\right), \quad (4.8)$$

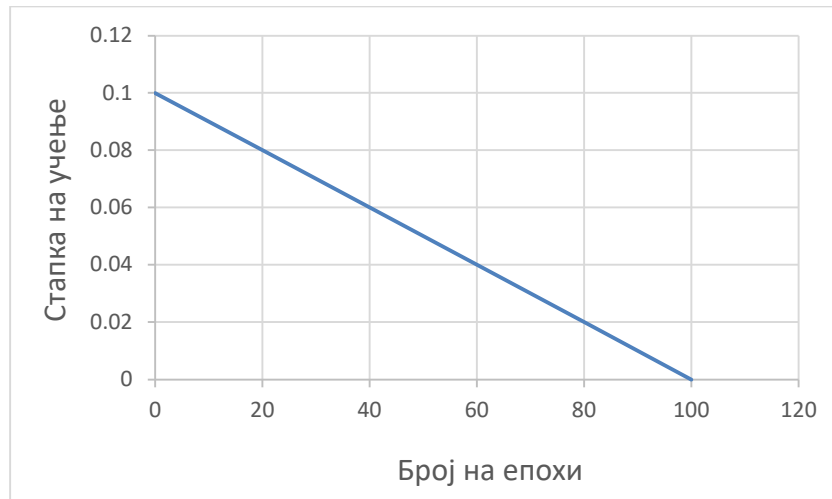
каде што T е бројот на тековната епоха и T_{max} е максималниот број на епохи.

Сите конволуциски невронски мрежи кои се користат во нашите експерименти за фино подесување се оригинално тренирани на ImageNet со различни распореди на опаѓање на стапките за учење: ResNet50 и DenseNet121 со опаѓање во скокови на стапката на учење и Inception V3 и Xception со експоненцијално опаѓање на стапката на учење.

4.9 Циклични стапки на учење

Цикличните стапки на учење (Cyclical Learning Rates (CLR)) ја елиминираат потребата да се идентификува оптималната вредност на почетната стапка на учење и распоредот на стапката на учење при тренирање на конволуциските мрежи (Smith 2017). Наспроти распоредите на стапките на учење, каде што стапката на учење константно се намалува, оваа техника

дозволува стапката на учење да осцилира во разумни граници. Цикличните стапки на учење даваат можност за поголема слобода при изборот на иницијалната вредност на стапката на учење. Тие водат кон побрза конвергенција при тренирање на невронските мрежи, со помал број на ажурирања на хиперпараметрите.

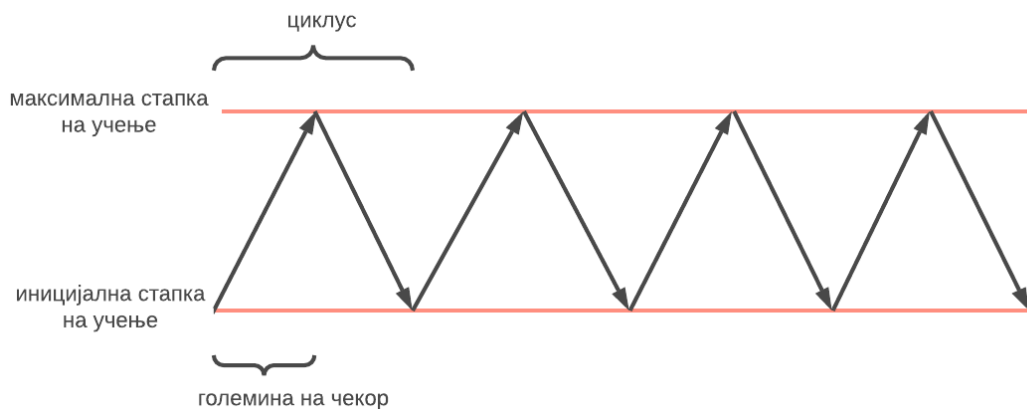


Слика 4.8 Линеарно опаѓање на стапката на учење при тренирање во 100 епохи
Figure 4.8. Linear learning rate decay applied to training in 100 epochs

„Точките на седло” (saddle points, превојни точки) се точки на критериумската функција (функцијата на загуба) каде што градиентот е нула, но тие не претставуваат минимум или максимум. Dauphin et al. (2015) откриле дека ефикасноста на методите со циклични стапки на учење лежи во топологијата на функцијата на загуба. Тие покажале дека превојните точки имаат полошо влијание на минимизирањето на функцијата на загуба отколку лошиот локален минимум. Процесот на учење се забавува поради малите градиенти на превојните точки. Кога стапката на учење се зголемува при осцилирање, тоа води кон надминување на проблемот со платоата на превојните точки. Друга причина за ефикасноста на методите со циклични стапки на учење е дека оптималната стапка на учење е некаде меѓу долната и горната граница, така што тренирањето се изведува со скоро оптимални стапки на учење.

Кај цикличните стапки на учење постојат три главни шеми: триаголна, како што е прикажано на сл. 4.9, триаголна 2 и експоненцијална. Триаголната шема е триаголен циклус: стапката на учење почнува од долната граница, ја зголемува вредноста до максималната за половина циклус и потоа се враќа на почетната

вредност на крајот на циклусот. Разликата меѓу триголната шема и триаголната 2 шема е дека кај втората горната граница на стапката на учење се намалува на пола после секој циклус. Тренирањето со триаголната 2 шема обезбедува постабилен тренинг. Експоненцијалната шема, како што сугерира и самото име, вклучува експоненцијално намалување на максималната стапка на учење (Smith 2017).



Слика 4.9 Циклична стапка на учење со триаголна шема

Figure 4.9 Cyclical learning rate with triangular policy mode

4.10 Измазнување на лабели

Измазнувањето на лабели (label smoothing) е регуларизациона техника која овозможува намалување на претренирањето (overfitting) и помага во подобрување на способноста на мрежните архитектури за генерализација. Измазнувањето на лабели е воведено од Szegedy et al. 2015, а е докажано дека ја зголемува точноста на класификација, усвојувајќи пондерирана сума на лабелите со униформна распределба наместо да се евалуира крос-ентропијата со „тврди“ лабели од податочното множество. Доделувањето на „тврди“ лабели кореспондира со бинарни лабели: позитивни за една класа и негативни за сите други класи. Доделувањето на „меки“ лабели ѝ дава најголема веројатност на позитивната класа и многу мали веројатности на другите класи. Измазнувањето на лабели се применува за да се спречи невронската мрежа да стане премногу сигурна во своите предвидувања. Со намалување на сигурноста на моделот, ние спречуваме тренирањето на мрежата да западне во длабоките долини на функцијата на загуба (Müller et al. 2019). Измазнувањето на лабели може исто така да се имплементира со додавање на негативната ентропија од софтмакс излезот на функцијата на веројатност при тренирањето, пондерирана со

дополнителен хиперпараметар (Pereyra et al. 2017; Guo et al. 2017; Goodfellow et al. 2016).

Предвидувањето на конволуциската мрежа е функција од активациите од второто до последното мрежно ниво:

$$y_k = \frac{e^{x^T w_k}}{\sum_{l=1}^L e^{x^T w_l}}, \quad (4.9)$$

каде што y_k е веројатност мрежата да класифицира во k –тата класа, тежините и праговите (bias) на последното мрежно ниво се дадени со w_k , x е векторот на активациите на претпоследното ниво споен со „1” за да се земе предвид прагот (bias). Ако се тренира мрежата со „тврди” лабели, намерата е да се минимизира крос-ентропијата меѓу реалните лабели t_k и предвидувањата на невронската мрежа y_k , како што следи:

$$E(t, y) = \sum_{k=1}^K -t_k \ln(y_k) \quad (4.10)$$

каде што t_k е „1” за точните лабели и „0” за останатите. Кога се тренира мрежата со измазнување на лабели со параметар α , се минимизира крос-ентропијата меѓу „измазнетите” лабели t_k^{LS} и предвидувањата на мрежата y_k . Измазнетите лабели се дадени со:

$$t_k^{LS} = t_k(1 - \alpha) + \alpha/K \quad (4.11)$$

Техниката на измазнување на лабели се користи кај предложениот метод со фино подесување со цел да се спречи конволуциската невронска мрежа да стане премногу сигурна во своите предвидувања и со тоа да се зголеми нејзината робустност и способност за предвидување.

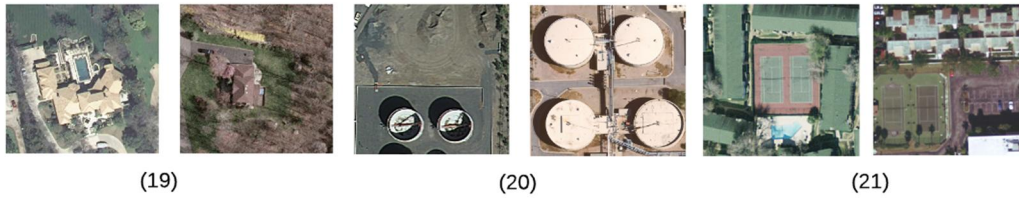
4.11 UC Merced податочно множество

UC Merced податочното множество (Yang et al. 2010) има 21 класа на воздушни сцени, како што се гледа на слика 4.10. Резолуцијата на пиксели на сликата е една стапка. Тие се исечени во региони од 256×256 пиксели. Оригиналните слики се преземени од Национална мапа на геолошки истражувања на САД (United States Geological Survey (USGS) National Map) од следните локации: Бирминген, Бостон, Бафало, Колумбус, Далас, Харисбург, Хјустон, Џексонвил, Лас Вегас, Лос Анџелес, Мајами, Напа, Њу Јорк, Рено, Сан Диого, Санта Барбара, Сиетл, Тампа, Туксон и Вентура. Секоја од класите има 100 слики, кои се мануелно селектирани и униформно обележани: (1)

земјоделско земјиште, (2) авион, (3) бејзбол дијамант, (4) плажа, (5) згради, (6) честар, (7) густа станбена област, (8) шума, (9) автопат, (10) терен за голф, (11) пристаниште, (12) крстосница, (13) станбена област со средна густина, (14) парк за мобилни куќички, (15) надвозник, (16) паркиралиште, (17) река, (18) писта, (19) ретка станбена област, (20) резервоари за складирање и (21) тениски терени.

Она што го прави податочното множество тешко за класификација е тоа дека постојат некои класи со сличен облик и дистрибуција на објектите, т.е. густа станбена област, станбена област со средна густина и ретка станбена област. Разликата меѓу овие класи е претежно во густината на објектите.

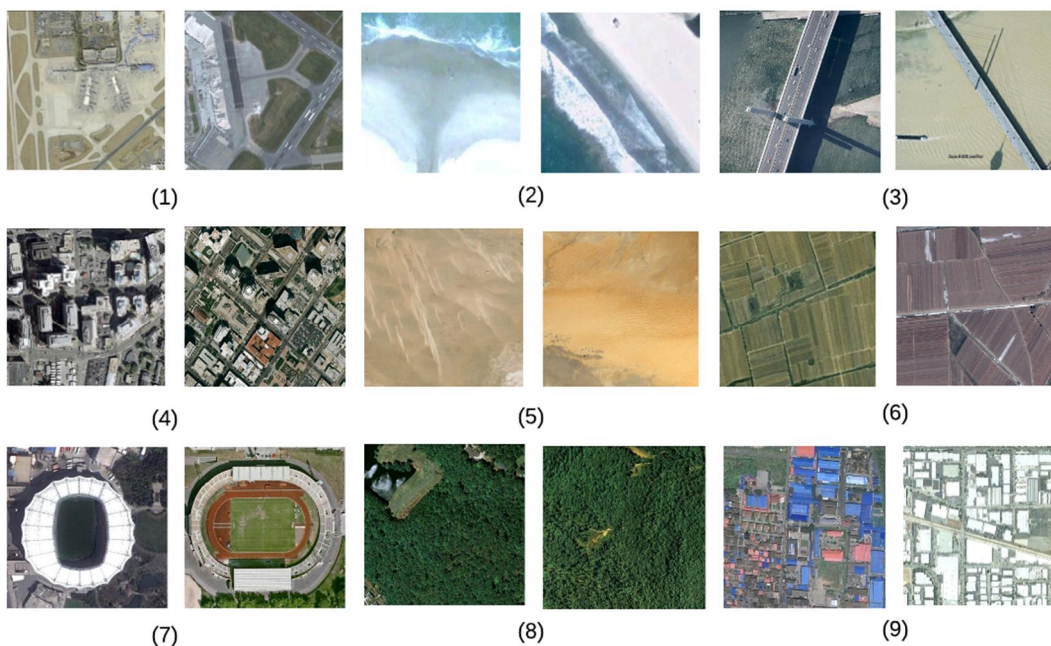




Слика 4.10 UC Merced податочно множество, 21 класа на слики
 Figure 4.10 UC Merced dataset, 21 image classes

4.12 WHU RS податочно множество

WHU RS податочното множество (Xia et al. 2010) е преземено од сликите на Google Earth. Има 1005 слики доделени во 19 класи: (1) аеродром, (2) плажа, (3) мост, (4) комерцијална област, (5) пустина, (6) земјоделско земјиште, (7) фудбалско игралиште, (8) шума, (9) индустриска област, (10) ливада, (11) планина, (12) парк, (13) паркиралиште, (14) езерце, (15) пристаниште, (16) железничка станица, (17) резиденцијална област, (18) река и (19) вијадукт. Сликите се со голема просторна резолуција и имаат 600 × 600 пиксели. Секоја класа е претставена од најмалку 50 слики. Примероци на слики од секоја класа се прикажани на сл. 4.11. Сликите во ова податочно множество претставуваат воздушни сцени од различни места од целата Земјина топка. Досега WHU RS податочното множество е екстензивно користено во истражувачки студии за различни методи на класификација на воздушни сцени.





Слика 4.11 WHU RS податочно множество, 19 класи на слики

Figure 4.11 WHU RS dataset, 19 image classes

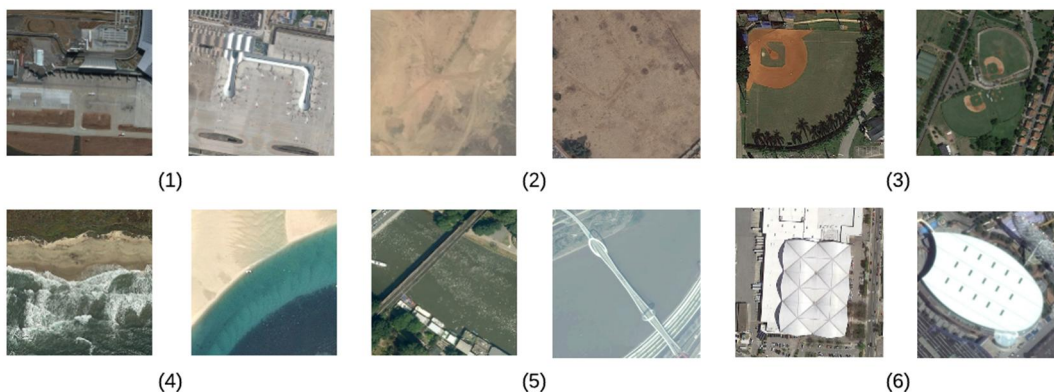
4.13 AID податочно множество

AID се состои од приближно 10,000 слики снимени од далечина со димензии 600 × 600 пиксели, распределени во 30 класи (Xia et al. 2017): (1) аеродром, (2) земјиште, (3) игралиште за бејзбол, (4) плажа, (5) мост, (6) центар, (7) црква, (8) комерцијална област, (9) густа станбена област, (10) пустина, (11) земјоделско земјиште, (12) шума, (13) индустриска област, (14) ливада, (15) средно населена станбена област, (16) планина, (17) парк, (18) паркиралиште, (19) игралиште, (20) езерце, (21) пристаниште, (22) железничка станица, (23) одморалиште, (24) река, (25) училиште, (26) ретка станбена област, (27) плоштад, (28) стадион, (29) резервоари за складирање, (30) вијадукт. Тие се преземени од Google Earth репозиториумот. Сликите се собрани од различни

континенти и држави во различни делови од годината и под различни временски услови: најмногу од Кина, Јапонија, Европа (Германија, Англија, Италија и Франција) и САД. Сликите од AID податочното множество имаат резолуција на пикселите од половина метар. Сликата 4.12 прикажува примероци на сликите од секоја класа.

4.14 NWPU-RESISC45 податочно множество

NWPU-RESISC45 податочното множество содржи слики собрани од Google Earth. Името на податочното множество доаѓа од името на универзитетот каде што е создадено (Northwestern Polytechnical University (NWPU)). Тоа содржи 31,500 слики снимени од воздух и поделени во 45 класи: (1) авион, (2) аеродром, (3) бејзбол дијамант, (4) кошаркарско игралиште, (5) плажа, (6) мост, (7) честар, (8) црква, (9) кружно земјоделско земјиште, (10) облак, (11) комерцијална област, (12) густа станбена област, (13) пустина, (14) шума, (15) автопат, (16) терен за голф, (17) атлетска патека, (18) пристаниште, (19) индустриска област, (20) крстосница, (21) остров, (22) езеро, (23) ливада, (24) средно населена станбена област, (25) парк за мобилни куќички, (26) планина, (27) надвозник, (28) замок, (29) паркиралиште, (30) пруга, (31) железничка станица, (32) правоаголно земјоделско земјиште, (33) река, (34) кружен тек, (35) писта, (36) санта лед, (37) брод, (38) снежен нанос, (39) ретка станбена област, (40) стадион, (41) резервоари за складирање, (42) тениско игралиште, (43) скалесто земјиште, (44) термоцентрала, (45) мочуриште. Секоја класа има 700 слики со димензии 256 × 256 пиксели. Освен четири класи (остров, езеро, планина и снежна покривка), кои покажуваат помала просторна резолуција, другите класи имаат просторни резолуции кои варираат во опсегот од 30m – 0.2m. Сликата 4.13 прикажува примероци на слики од секоја класа.





Слика 4.12 AID податочно множество, 30 класи на слики
 Figure 4.12 AID dataset, 30 image classes





Слика 4.13 NWPU-RESISC45 податочно множество, 45 класи на слики
 Figure 4.13 NWPU-RESISC45 dataset, 45 image classes

4.15 Експериментална поставка за предложениот метод со екстракција на карактеристики

Во сите експерименти опфатени со предложениот метод со екстракција на карактеристики, се користи умерено вештачко проширување на множеството за тренирање (data augmentation). При тоа се креираат 5 примероци од една слика, со операции како транслации, ротации, промени во размерот, отсекување на делови од сликата и хоризонтални и вертикални превртувања на сликите од податочното множество. При поагресивно проширување на множеството за тренирање може да се генерираат и повеќе примероци од оригиналната слика.

Нашите симулации се водат во две насоки. Прво, се екстрахираат карактеристики од три различни слоеви кај секоја од конволуциските мрежи: ResNet50, InceptionV3, Xception и DenseNet121. Овие конволуциски мрежи се

претходно тренирани и параметрите на оригиналните длабоки архитектури се задржани. Кај мрежата ResNet50 се користат следните слоеви: нивото за редукција на просек, последното конволуциско ниво пред нивото за редукција и конволуциското ниво со ознака bn4f_branch2c. Кај конволуциската мрежа InceptionV3 се користат слоевите: нивото за редукција на просек и конволуциските нивоа mixed_10 и mixed_8. Кај Xception предмет на анализа се следните слоеви: нивото за редукција на просек и двете конволуциски нивоа block14_sepconv2_act и block14_sepconv1_act. Конечно, за DenseNet121 мрежата се користат следните нивоа: нивото за редукција на просек, последното конволуциско ниво пред нивото за редукција и conv4_block24_concat конволуциското ниво. Димензиите на влезните слики за ResNet50 и DenseNet121 се 224×224 , а димензиите на влезните слики за InceptionV3 и Xception се 229×299 . Екстракцијата на карактеристики се изведува за две податочни множества: UC Merced и WHU RS. Соодносот на подмножествата за тренирање и тестирање е 80% наспроти 20% за UC Merced податочното множество и 60% наспроти 40% за WHU RS податочното множество. Одбраните соодноси се исти како и оние во трудовите со кои го споредуваме нашиот пристап. Поделбата е случајна. Не се користи стратификација. Не користиме стратификација при поделбата на податоците за тренирање и тестирање со цел да имаме комплетно случаен процес. Контролираната поделба на податоци (со еднаков број на слики од секоја класа за тренирање и тестирање при секоја поделба на податочното множество) може да влијае на точноста на класификација и да води кон повисока средна вредност и помала стандардна девијација на резултатите. Сепак, целта не е да се избегнат најлошите резултати при класификација. Секоја влезна слика претходно се обработува согласно барањата на соодветната конволуциска мрежа. Се користи умерено вештачко проширување на подмножеството за тренирање (генераторот на слики произведува пет примероци по слика): сликите се ротираат, се поместуваат, им се отсекуват делови, се зумираат и хоризонтално се превртуваат. Откако ќе се екстрахираат карактеристиките, се тренира линеарен класификатор (LRC или SVM). Се изведува мрежно пребарување со цел да се подесат хиперпараметрите на моделот (регуларизациониот параметар C). Првиот дел од истражувањето се спроведува со единствена цел да се добие претстава за мрежните нивоа кај конволуциските мрежи од кои екстракцијата на карактеристики води кон најдобри резултати при

класификација. Тука, важно е да се напомене дека нивото за редукција на просек или последното конволуциско ниво не секогаш даваат највисока точност на класификација. Кај некои мрежи, најдобри резултати се добиваат со употреба на внатрешните (средни) конволуциски нивоа.

Вториот дел од истражувањето поврзано со методот со екстракција на карактеристики се води за подобрување на резултатите при класификација со користење на фузија на карактеристики. Овој пат, карактеристиките се екстрахираат од две различни нивоа од две различни конволуциски мрежи, на таков начин што едното ниво е секогаш она за редукција на просек (average pooling), а другото ниво е конволуциско ниво (последното или некое од средните конволуциски нивоа). Комбинациите на нивоа и мрежи се одредуваат врз основа на резултатите добиени од првиот дел на истражувањето. Тука, сликите се повторно соодветно зголемени/ намалени, претходно се обработени и повторно е применето умерено вештачко проширување на подмножеството за тренирање: сликите се ротираат, поместуваат, им се отсекуваат делови, се зумираат и хоризонтално се превртуваат. Големинскиот сооднос на подмножествата за тренирање и тестирање е 80%/20% и 50%/50% за UC Merced податочното множество и 60%/40% и 40%/60% за WHU RS податочното множество. Поделбата е случајна и повторно без стратификација. Пред фузијата на карактеристики (спојување), се изведува трансформација со анализа на главните компоненти (PCA) на екстрахираните карактеристики од конволуциското ниво. Потоа, се прави L2 нормализација на екстрахираните карактеристики од нивото за редукција на просек и трансформираните карактеристики и конечно, карактеристиките се фузираат. За задачата на класификација се применува машина со вектори за поддршка (SVM). Добиените експериментални резултати со овој метод се споредливи со најсовремените методи базирани на екстракција на карактеристики.

Сите симулации се изведуваат на оперативен систем Ubuntu 18.04 со Keras верзија 2.2.4. Google библиотеката TensorFlow верзија 1.12.0 (Abadi et al. 2016) е во позадина на Keras. Хардверската конфигурација е: CPU i7-8700 3.2 GHz и 64 GB RAM. Графичкиот процесор е Nvidia GeForce GTX 1080 Ti, со 11 GB сопствена меморија и CUDA верзија 9.0 инсталирано на неа.

4.16 Експериментална поставка за предложениот метод со фино подесување

Нашиот предложен метод кој користи фино подесување како форма на пренос на учење, се изведува со линеарно опаѓање на стапката на учење и циклични стапки на учење, како и со измазнување на лабели за класификација на сцени снимени од воздух. Во експериментите ги користиме истите конволуциски невронски мрежи претходно тренирани на ImageNet податочното множество како и кај методот со екстракција на карактеристики: ResNet50, InceptionV3, Xception и DenseNet121. Финото подесување се изведува со „мрежна операција“, т.е. со отстранување на последните слоеви кај секоја од претходно тренираните мрежи по нивото за редукација на просек. По ова се конструира нова глава на мрежата со додавање на целосно поврзано ниво, ниво за испуштање и софтмакс (softmax) ниво за класификација, кое содржи број на излезни неврони кој соодветствува на бројот на класи кај испитуваните податочни множества.

Како што е веќе споменато, кај методот за класификација на слики снимени од далечина со фино подесување се анализираат две податочни множества од поголем обем: AID и NWPU-RESISC45. Сликите од податочните множества се намалени, односно зголемени согласно барањата на соодветните конволуциски мрежи: 224×224 пиксели за ResNet50 и DenseNet121, како и 299×299 пиксели за InceptionV3 и Xception. Експериментите се изведуваат согласно следните односи на обемот на подмножествата за тренирање и тестирање: 50%/50% и 20%/80% за AID податочното множество и 20%/80% и 10%/90% за NWPU-RESISC45 податочното множество. Избраните размери на поделба кореспондираат со оние во научните трудови од истата област, со кои го споредуваме нашиот пристап. За сликите од подмножеството за тренирање се користи вештачко проширување (Shorten et al. 2019). Ова е регуларизациона техника која го зголемува обемот на податочното множество и скоро секогаш резултира со зголемување на точноста на класификација. Уште повеќе, во сите експерименти се вклучува и регуларизационата техника измазнување на лабели. Измазнувањето на лабели, како и вештачкото проширување, се применува само на сликите за тренирање. Тоа резултира во поголеми загуби при тренирање споредено со загубите при тестирање. Спротивно на ова, измазнувањето на лабели (label smoothing) превенира претренирање (overfitting)

и помага нашиот модел да генерализира подобро. Претренирањето е вообичаен проблем кога се користат конволуциски мрежи со голема димензионалност, кои се претходно тренирани на множества со милиони слики, да решаваат задачи на класификација за податочни множества кои содржат неколку илјади слики.

Првиот дел од процесот на фино подесување почнува со загревање на новите мрежни нивоа од главата на конволуциската мрежа. Тежините на слоевите од новата мрежна глава случајно се иницијализираат. Но, другите мрежни нивоа по мрежната операција ги задржуваат тежините од првобитното тренирање. Значи, потребно е слоевите на новата мрежна глава да почнат да го учат податочното множество кое е предмет на класификација (да се „загреат“). За време на процесот на загревање, единствени нивоа кои се тренираат се оние од новата мрежна глава, другите мрежни слоеви се замрзнати. Загревањето на новата мрежна глава се прави со RMSprop алгоритмот и константна стапка на учење. Финото подесување на мрежниот модел продолжува со стохастички градиентен спуст (Stochastic Gradient Descent (SGD)) и овој пат сите мрежни нивоа се „одмрзнуваат“ за тренирање. Посебни експерименти се спроведуваат за линеарно опаѓање на стапката на учење и за циклични стапки на учење со триаголна шема. Триаголната шема е избрана од причини што таа е најшироко користена во литературата и со неа се добива највисока точност на класификација, споредено со другите шеми на цикличните стапки на учење (Cyclical Learning Rate (CLR)). Кога се применува линеарно опаѓање на стапката на учење, истата константно се намалува до нула на крајот на последната епоха на тренирање. Најголемиот предизвик во овој случај е да се избере почетната стапка на учење, која се зема 1-2 реда на величина помала од стапката на учење со која била тренирана оригиналната мрежа. Во однос на цикличните стапки на учење, ја осцилираме стапката на учење меѓу минималната и максималната вредност, претпоставувајќи дека оптималната стапка на учење е некаде во интервалот. Изборот на долна и горна граница на цикличните стапки на учење не е толку чувствителен како изборот на почетна стапка на учење кај линеарното опаѓање на стапките на учење. Тука користиме големина на чекорот 4 или 8 пати по бројот на итерации на тренирање во епохата. Бројот на епохите на тренирање се одредува за да содржи цел број на циклуси. Ова се прави за да се задржи идејата која лежи во основата на цикличните стапки на учење: да се почне од една минимална вредност на стапката на учење, потоа таа се зголемува до

максималната вредност и на крајот повторно се враќаеме на почетната стапка на учење. Со ова сме завршиле еден циклус и повторно се почнува од почеток.

Вториот дел од истражувањето на предложениот метод со фино подесување е посветен на евалуацијата на методите на класификација, односно софтмакс класификатор и машина со вектори за поддршка со линеарно и Гаусово јадро. По финото подесување на секоја конволуциска мрежа, се пресметува точноста на класификација со софтмакс нивото, кое е дел од новата мрежна глава и е тренирано заедно со сите други мрежни слоеви. За да се споредат резултатите добиени со софтмакс класификаторот со двата типа на машината со вектори за поддршка, фино подесените конволуциски невронски мрежи се користат како екстрактори на карактеристики. Се екстрахираат сликовите карактеристики од двете податочни множества на слики снимени од далечина од целосно поврзаното ниво на фино подесените мрежи. Потоа, екстрахираните карактеристики се користат за тренирање на линеарна и Гаусова машина со вектори за поддршка и за класифицирање на сликите од податочните множества. Класификацијата се изведува за сите размери на поделба на податочните множества, со примена и на линеарно опаѓање на стапката на учење и на циклични стапки на учење и измазнување на лабели во секое симулационо сценарио.

Како и кај предложениот метод со екстракција на карактеристики и овде експериментите се изведуваат на Ubuntu 18.04 оперативен систем и Keras верзија 2.2.4. Во позадината на Keras е библиотеката на Google, TensorFlow верзија 1.12.0. Хардверската конфигурација е: CPU i7-8700 3.2 GHz и 64 GB RAM. Графичката картица е Nvidia GeForce GTX 1080 Ti, со 11 GB меморија и инсталирано CUDA верзија 9.0.

4.17 Мерки за евалуација

Во овој докторски труд се користат две мерки за евалуација: вкупна точност (Overall Accuracy (OA)), која е еквивалент на точноста на класификација и матрица на конфузија. Овие две мерки се вообичаени за анализа на резултатите и споредба со слични современи методи во научните истражувања на проблеми на класификација. Точност на класификација е однос меѓу бројот на соодветно класифицирани слики за тестирање (од сите класи на слики) и вкупниот број на слики за тестирање. Вредноста на оваа мерка е помалку од 1.

Наспроти вкупната точност, точноста на класификација за секоја класа на слики посебно се презентира во матрица на конфузија. Матрицата на конфузија е графички приказ (табела) за точноста на класификација за секоја класа од податочното множество. Оваа табела јасно ги покажува грешките за секоја класа посебно и конфузијата меѓу различни класи. Во матрицата на конфузија, колоните ги претставуваат предвидените класи, додека редовите ги претставуваат вистинските класи. Кај нормализираната матрица на конфузија членот x_{ij} претставува процент на слики кои се класифицирани како да припаѓаат на i -тата класа, но нивната вистинска класа е j (Zhang W. et al. 2019). Идеално прецизен модел на предвидување води кон дијагонална матрица на конфузија, или кон матрица со големи вредности на дијагоналата и мали вредности за останатите членови. Во нашите експериментални постановки, податочните множества се поделени на дел за тренирање и дел за тестирање. Поделбите се случајни, без стратификација и се направени согласно соодносите наведени во претходните подпоглавја. Со цел да се провери сигурноста на резултатите, одредени експериментални сценарија се повторуваат повеќе пати. Кај предложениот метод со екстракција на карактеристики, сите случаи каде што се добива висока точност на класификација се повторуваат десет пати. Додека кај техниката со фино подесување, по пет пати се повторуваат експерименталните сценарија на класификација со софтмакс класификатор при тренирање „од крај до крај“ (end-to-end) и соодвето при класификација на екстрахираните сликови карактеристики со линеарна машина со вектори за поддршка. Во сите случаи на повеќекратно повторување на симулациите, се пресметува средна вредност и стандардна девијација на точноста на класификација.

5. РЕЗУЛТАТИ

5.1 Резултати од методот со екстракција на карактеристики

5.1.1 Класификација базирана на екстрахирани карактеристики од различни мрежни нивоа

Како што е објаснето во секцијата 4.15, првиот дел од експерименталните испитувања на предложениот метод со екстракција на карактеристики е посветен на класификација базирана на екстрахирани карактеристики единствено од едно мрежно ниво. Добиените резултати се прикажани во табелата 5.1, со 80% од UC Merced податочното множество наменето за тренирање и во табелата 5.2 за WHU RS податочното множество со 60% од неговиот обем наменет за тренирање. Преостанатите слики од податочните множества се користат за тестирање и не се двои дел од множествата за валидација. Главната причина поради која не се користи подмножество за валидација лежи во фактот дека множествата се мали по обем и не е оправдано понатаму да се намалува бројот на слики за тренирање и тестирање. Експериментите се изведуваат екстензивно со сите четири претходно тренирани конволуциски мрежи и три различни слоја кај секоја од нив. Екстрахираните карактеристики се водат на два различни класификатора: класификатор со логистичка регресија и машина со вектори за поддршка со линеарно јадро.

Како што е познато, нивото за редукција на просек е замена за целосно поврзаните нивоа од архитектурата на конволуциските мрежи. Од тие причини се очекува дека екстрахираните карактеристики од ова ниво би дале најголема точност на класификација. Од друга страна конволуциските нивоа, посебно средните, даваат карактеристики кои претставуваат информации од средно ниво (пр. делови од објекти), но не и просторната зависност меѓу нив (пр. цели објекти) и точноста на класификација би требало да е полоша споредено со карактеристиките извлечени од слоевите за редукција на просек.

Меѓутоа ако внимателно се анализираат табелите 5.1 и 5.2, може да се заклучи за двете податочни множества, како и за двата линеарни класификатора, дека најголема точност на класификација за секоја од претходно тренираните мрежи, освен за ResNet50, се добива со карактеристиките извлечени од средниот конволуциски слој. Тоа е mixed_8 нивото за InceptionV3, block14_sepconv1_act нивото за Xception и conv4_block24_concat нивото за DenseNet121. Овие резултати ни даваат насоки за вториот дел од

експериментите за предложената техника со екстракција на карактеристики. Таму екстрахираме карактеристики од конволуциските нивоа од интерес, потоа ги трансформираме со анализа на главните компоненти (Principal Component Analysis (PCA)) и истите се фузираат со екстрахираните карактеристики од нивото за редуција на просек од различни мрежи. Со предложената техника се цели кон подобрување на точноста на класификација на UC Merced и WHU RS податочното множество.

Табела 5.1 Точност на класификација (OA (%)) со употреба на карактеристики од различни мрежни нивоа со 80% од UC Merced наменети за тренирање

Table 5.1 The classification accuracy (OA (%)) using features from different network layers with 80% of UC Merced dataset as a training set

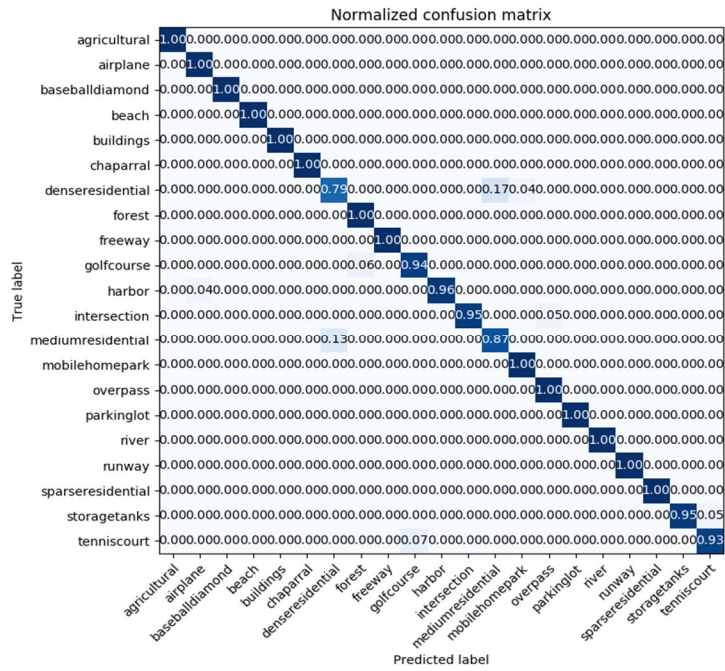
Метод/ Method	класификатор со логистичка регресија/ LRC	машина со вектори за поддршка/ SVM
ResNet50		
ниво за редуција на просек/ avg pooling	96.19	95.71
последно конволуциско ниво/ last conv layer	95.71	97.38
bn4f_branch2c	94.52	93.57
InceptionV3		
ниво за редуција на просек/ avg pooling	96.67	95
mixed_10	95.48	95.71
mixed_8	98.10	98.33
Xception		
ниво за редуција на просек/ avg pooling	93.57	94.76
block14_sepconv2_act	93.81	94.29
block14_sepconv1_act	96.43	95.71
DenseNet121		
ниво за редуција на просек/ avg pooling	95.48	93.81
conv5_block16_concat	96.67	94.05
conv4_block24_concat	97.14	95.24

Table 5.2 Точност на класификација (OA (%)) со употреба на карактеристики од различни мрежни нивоа со 60% од WHU RS наменети за тренирање

Table 5.2 The classification accuracy (OA (%)) using features from different layers with 60% of WHU RS dataset as a training set

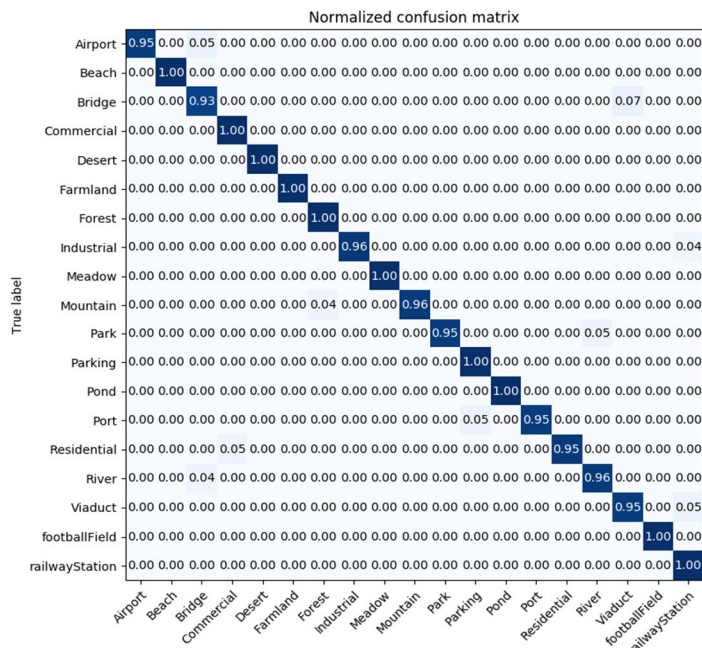
Метод/ Method	класифкатор со логистичка регресија/ LRC	машина со вектори за поддршка/ SVM
ResNet50		
ниво за редукација на просек/ avg pooling	98.01	97.01
последно конволуциско ниво/ last conv layer	98.01	97.76
bn4f_branch2c	95.52	96.02
InceptionV3		
ниво за редукација на просек/ avg pooling	95.78	95.02
mixed_10	94.53	95.52
mixed_8	97.26	97.26
Xception		
ниво за редукација на просек/ avg pooling	93.28	93.53
block14_sepconv2_act	94.28	94.53
block14_sepconv1_act	95.27	95.52
DenseNet121		
ниво за редукација на просек/ avg pooling	96.52	95.27
conv5_block16_concat	96.27	95.52
conv4_block24_concat	96.27	96.27

Во табелите 5.1 и 5.2 се дадени вкупните постигнати точности на класификација (OA) за сите класи од разгледуваните податочни множества. Парцијалните точности на класификација, за секоја класа одделно, графички се прикажуваат со матрици на конфузија. Сликата 5.1 прикажува матрици на конфузија за класификација на UC Merced податочното множество при 80% обем на сликите за тренирање. Сликата 5.2 прикажува матрици на конфузија за класификација на WHU RS податочното множество при 60% обем на сликите за тренирање.

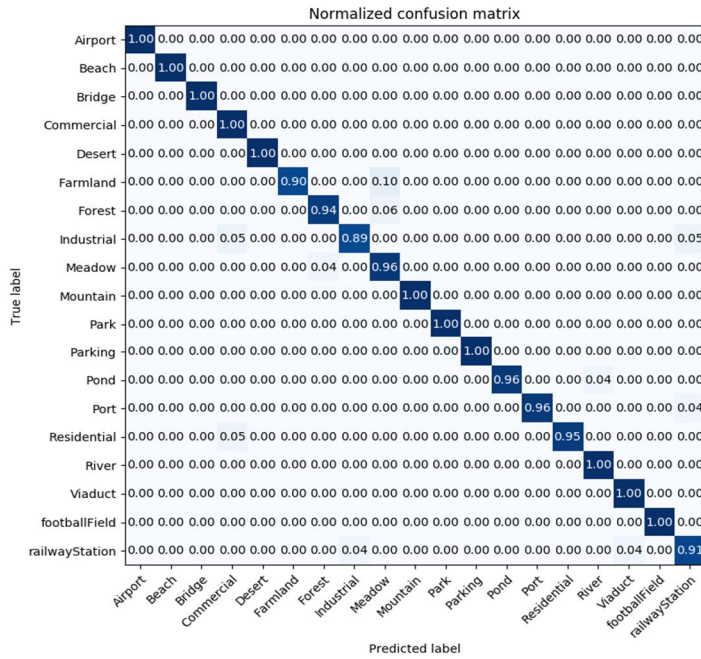


(3)

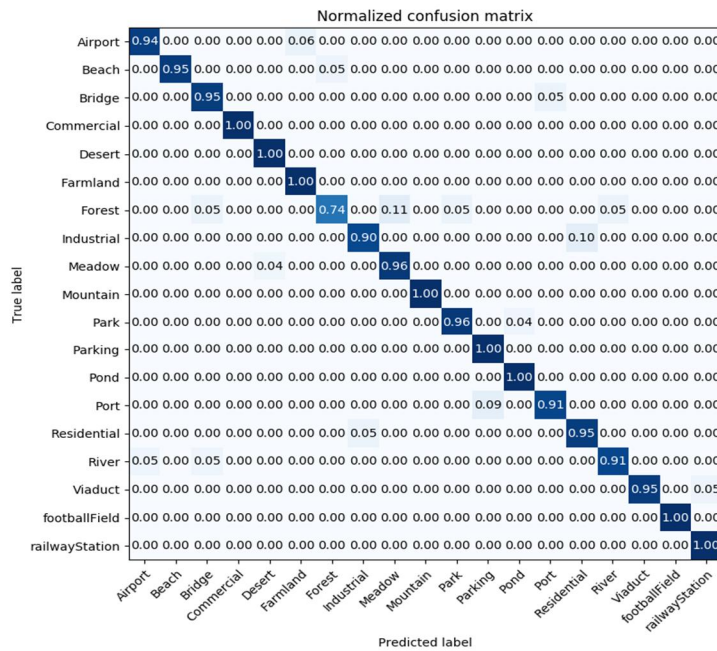
Слика 5.1 Матрици на конфузија за 80% обем за тренирање од UC Merced за (1) ResNet50 ниво за редукција на просек; (2) Xception block14_sepconv1_act; (3) DenseNet121 conv4_block24_concat и класификатор со логистичка регресија
 Figure 5.1 The confusion matrices under 80% training ratio of the UC Merced for (1) ResNet50 average pooling; (2) Xception block14_sepconv1_act; (3) DenseNet121 conv4_block24_concat and LRC



(1)



(2)



(3)

Слика 5.2 Матрици на конфузија за 60% обем за тренирање од WHU RS за (1) ResNet50 последно коновлуциско ниво; (2) InceptionV3 mixed_8; (3) DenseNet121 conv5_block16_concat и машина со вектори за поддршка

Figure 5.2 The confusion matrices of classification under 60% training ratio of the WHU RS for (1) ResNet50 last conv; (2) InceptionV3 mixed_8; (3) DenseNet121 conv5_block16_concat and SVM

5.1.2 Класификација базирана на фузија на карактеристики трансформирани со анализа на главните компоненти

Вториот дел од симулациите го покажуваат капацитетот на евалуираниот метод со екстракција на карактеристики: линеарна класификација на карактеристиките споени од нивото за редукција на просек и трансформираниите карактеристики со анализа на главните компоненти од некое од конволуциските нивоа. При декомпозицијата (трансформацијата), бројот на компоненти е од редот на величина на бројот на карактеристики од нивоата за редукција, што обезбедува добар баланс меѓу перформансите и редукцијата на димензионалност. За линеарната машина со вектори за поддршка се користи мрежно пребарување (Lameski et al. 2015) за да се одбере вредност за C од множеството на вредности: 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10.000 и 100.000. Максималниот број на итерации е 5000, со употреба на 3-кратна стандардна крос-валидација на подмножеството за тренирање. Стандардните параметри на машината со вектори за поддршка се значително полоши без мрежно пребарување.

Истражувањето е направено за различни комбинации на слоеви и машина со вектори за поддршка како линеарен класификатор за UC Merced податочното множество, па е презентирано во табелата 5.3. За да се евалуира точноста на класификација на истражуваната техника, ја споредуваме со постигнатата точност на класификација со неколку слични современи техники за класификација на UC Merced податочното множество. Перформансите на различните техники се дадени во табелата 5.4, како вообичаен начин на прикажување на резултати со користење на овие методи (Yu et al. 2018a). Како што покажува табелата 5.4, предложениот метод со фузија на карактеристики со трансформација на главните компоненти дава точност на класификација споредлива со конкурентните методи. Можат да се најдат методи кои го надминуваат нашиот модел: интеграција на карактеристики со глобален контекст и карактеристики кои се однесуваат на локалните објекти (Global Context Features + Local Object Features (GCF+LOF)) (Zeng et al. 2018) и Inception-v3-CapsNet (Zhang W. et al. 2019). Со цел да се провери сигурноста на резултатите, случаите каде што се добива најголема точност на класификација се повторуваат десет пати, а потоа се пресметува средна вредност и стандардна

девијација на точноста на класификација, како што е дадено во табела 5.5. Може да се забележи дека усреднетите точности се нешто пониски од оние прикажани во табелата 5.4.

Табела 5.3 Точност на класификација (ОА (%)) на UC Merced за фузирани карактеристики со трансформација на главните компоненти

Table 5.3 The classification accuracy (OA (%)) of UC Merced dataset for fused features with PCA transformation

Метод/ Method	80% од UCM/ 80% of UCM	50% од UCM/ 50% of UCM
ResNet50 последно конволуциско ниво (last conv layer) + InceptionV3 ниво за редуција (avg pooling)	97.14	97.33
ResNet50 последно конволуциско ниво (last conv layer) + Xception ниво за редуција (avg pooling)	97.62	97.43
DenseNet121 conv5_block16_concat + Xception ниво за редуција (avg pooling)	97.86	96.67
DenseNet121 conv4_block24_concat + Xception ниво за редуција (avg pooling)	97.86	96.57
InceptionV3 mixed_10 + ResNet50 ниво за редуција (avg pooling)	97.62	96.57
InceptionV3 mixed_8 + ResNet50 ниво за редуција (avg pooling)	98.33	97.43
InceptionV3 mixed_10 + Xception ниво за редуција (avg pooling)	95.95	95.14
InceptionV3 mixed_8 + Xception ниво за редуција (avg pooling)	98.57	97.62
DenseNet121 conv5_block16_concat + ResNet50 ниво за редуција (avg pooling)	97.14	96.67
DenseNet121 conv4_block24_concat + ResNet50 ниво за редуција (avg pooling)	96.9	95.24
Xception block14_sepconv2_act + DenseNet121 ниво за редуција (avg pooling)	96.67	96.48
Xception block14_sepconv1_act + DenseNet121 ниво за редуција (avg pooling)	98.57	96.29

Табела 5.4 Точност на класификација (OA (%) и SD) за методот со фузија на карактеристики и референтните методи за UC Merced

Table 5.4 The classification accuracy (OA (%) and SD) of the feature fusion method and the reference methods for UC Merced dataset

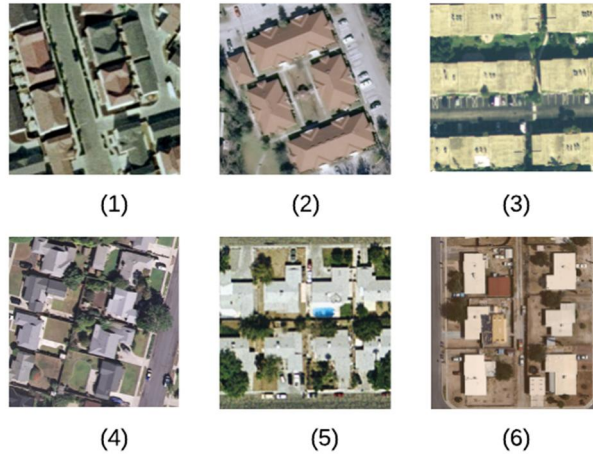
Метод/ Method	80% од UCM/ 80% of UCM	50% од UCM/ 50% of UCM
CaffeNet (Xia et al. 2017)	95.02 ± 0.81	93.98 ± 0.67
GoogLeNet (Xia et al. 2017)	94.31 ± 0.89	92.70 ± 0.60
VGG-16 (Xia et al. 2017)	95.21 ± 1.20	94.14 ± 0.69
„растегната“ конволуциска невронска мрежа со случаен размер/ SRSCNN (Liu et al. 2018a)	95.57	/
конволуциска мрежа – машина за екстремно учење/ CNN-ELM (Chen J. et al. 2018)	95.62	/
salM ³ LBP-CLM (Bian et al. 2018)	95.75 ± 0.80	94.21 ± 0.75
TEX-Net-LF (Anwer et al. 2017)	96.62 ± 0.49	95.89 ± 0.37
вреќа на визуелни зборови со локални и глобални карактеристики/ LGFBOVW (Zhu et al. 2016)	96.88 ± 1.32	/
фино подесена GoogLeNet/ fine-tuned GoogLeNet (Castellucio et al. 2015)	97.10	/
фузија со додавање/ fusion by addition (Chaib et al. 2017)	97.42 ± 1.79	/
CCP-net (Qi et al. 2018)	97.52 ± 0.97	/
фузија на два извора/ two-stream fusion (Yu et al. 2018b)	98.02 ± 1.03	96.97 ± 0.75
DSFATN (Gong et al. 2018)	98.25	/
пренос од длабоки конволуциски мрежи/ deep CNN transfer (Hu et al. 2015)	98.49	/
InceptionV3 mixed_8 + Xception ниво за редукција (avg pooling) (предложениот метод)	98.57	97.62
Карактеристики со глобален контекст + карактеристики кои се однесуваат на локалните објекти /GCFs+LOFs (Zeng et al. 2018)	99 ± 0.35	97.37 ± 0.44
InceptionV3-CapsNet (Zhang W. et al. 2019)	99.05 ± 0.24	97.59 ± 0.16

Табела 5.5 Вкупна точност (%) и стандардна девијација за предложениот метод со фузија на карактеристики за UC Merced

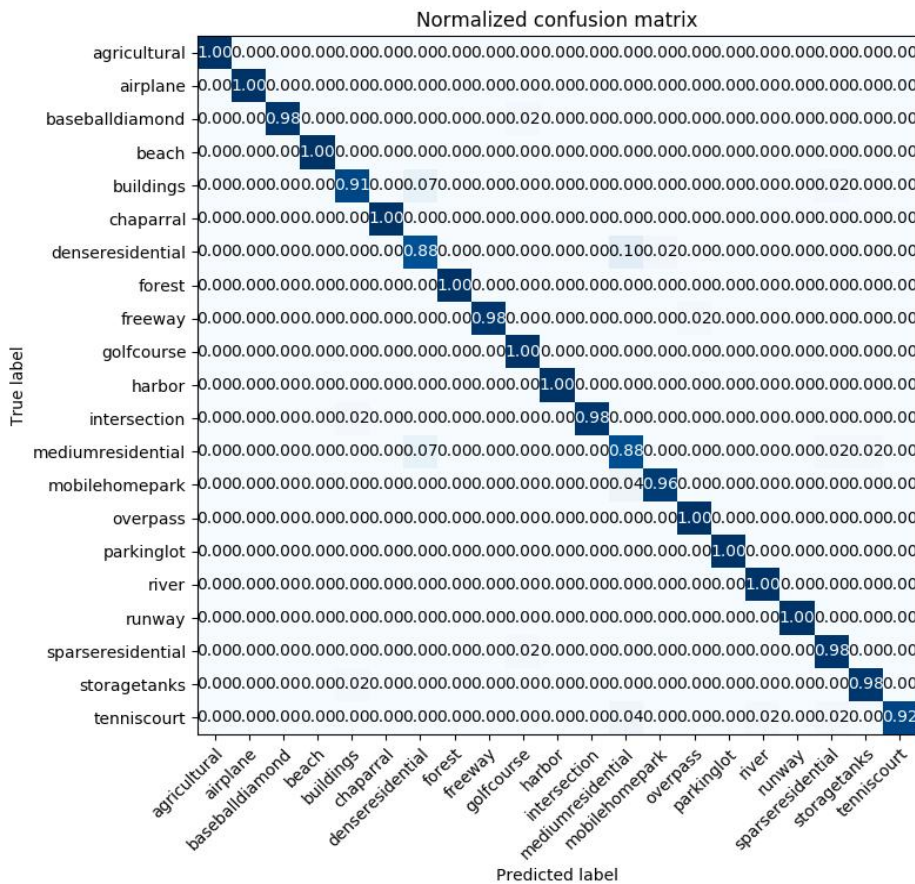
Table 5.5 The overall accuracy (%) and standard deviation of the proposed feature fusion method for UC Merced dataset

Метод/ Method	80% од UCM/ 80% of UCM	50% од UCM/ 50% of UCM
InceptionV3 mixed_8 + ResNet50 ниво за редукција (avg pooling)	97.67 ± 0.64	97.00 ± 0.65
InceptionV3 mixed_8 + Xception ниво за редукција (avg pooling)	97.86 ± 0.59	96.6 ± 0.65

Сликата 5.4 ја прикажува матрицата на конфузија во случајот со најдобра точност на класификација, добиена со комбинација од трансформираниите карактеристики од InceptionV3 mixed_8 и Xception нивото за редукција на просек со 50% од UC Merced податочното множество наменето за тренирање. Матрицата на конфузија покажува дека најлошо се класифицирани категориите „густа станбена област“ и „станбена област со средна густина“. Овие две класи лесно се мешаат една со друга. Ова доаѓа од фактот што класите „густа станбена област“ и „станбена област со средна густина“ имаат слична структура на сликите, на пр. облик и распределба на зградите, па затоа е тешко да се разликуваат една од друга, што може да се види на сликата 5.3. Се забележува дека класата „густа станбена област“, како и „станбена област со средна густина“, постигнале точност од 88%. Оваа точност го надминува методот базиран на комбинација на карактеристики со глобален контекст и карактеристики кои се однесуваат на локалните објекти (Zeng et al. 2018), со точност на категоријата „густа станбена област“ од 74% и методот Inception-v3-CapsNet (Zhang W. et al. 2019), со точност на истата категорија од 80%. Овие две техники се најдобро рангирани во табелата 5.4.

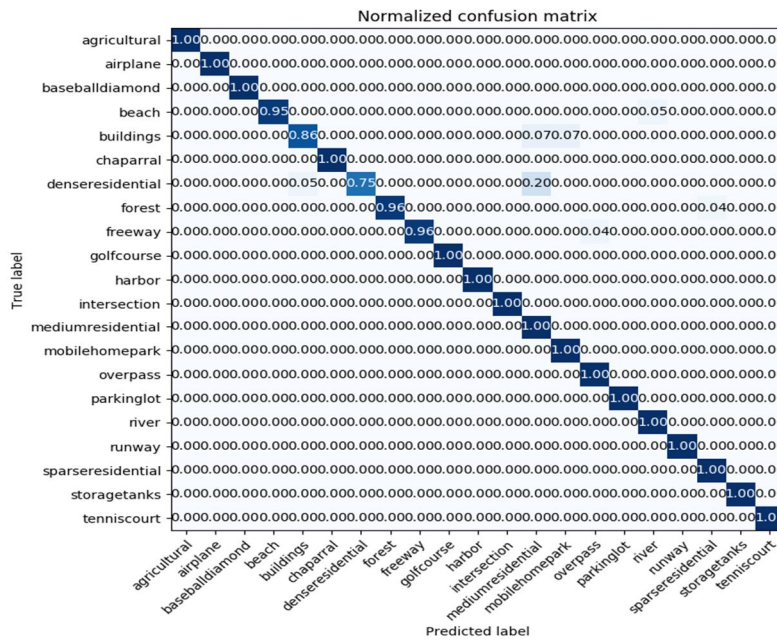


Слика 5.3 Претставници на класите од UC Merced LandUse: (1), (2), (3) густа станбена област; (4), (5), (6) станбена област со средна густина
 Figure 5.3 Class representatives of the UC Merced LandUse dataset: (1), (2), (3) dense residential; (4), (5), (6) medium residential

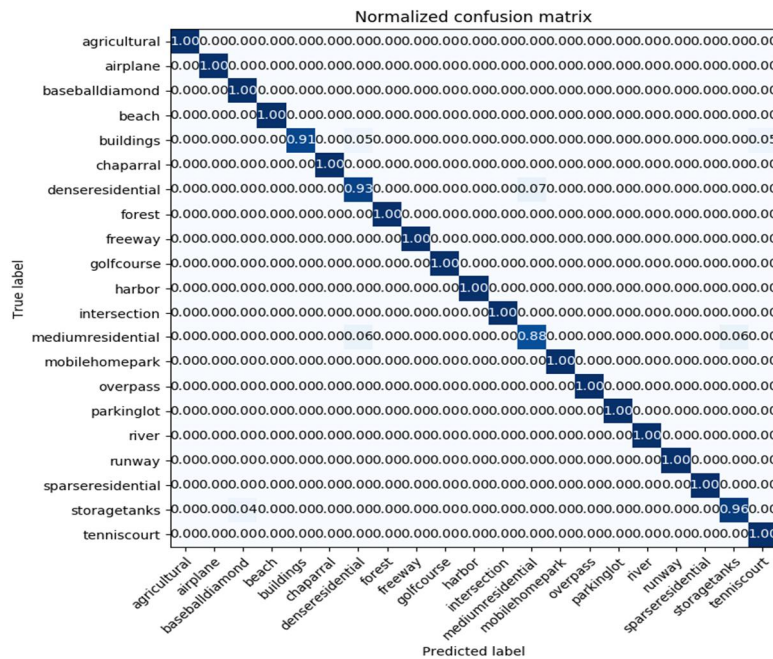


Слика 5.4 Матрица на конфузија на фузирани карактеристики, со 50% од UC Merced за тренирање со InceptionV3 mixed_8 + Xception ниво за редукција
 Figure 5.4 The confusion matrix of the fused features, with 50% of the UC Merced dataset as a training set for InceptionV3 mixed_8 + Xception avg pooling

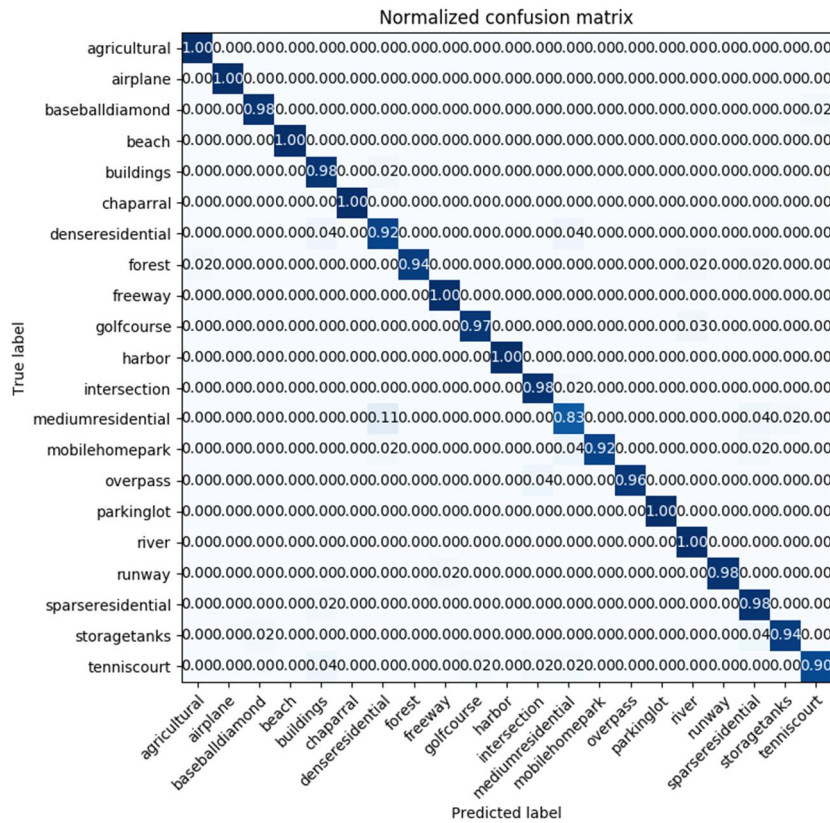
Во продолжение се дадени матриците на конфузија кои се генерирани при класификација на UC Merced податочното множество со помош на споени карактеристики од различни слоеви и различни конволуциски невронски мрежи. Овие матрици, како и сите досега прикажани, се нормализирани, т.е. членовите на матриците ја даваат процентуалната точност на класификација по категории и се помали од 1.



(1)



(2)



(5)

Слика 5.5 Матрици на конфузија за испитуваниот метод со 80% од UC Merced податочното множество за тренирање за (1) InceptionV3 mixed_8 + ResNet50 ниво за редукација, (2) InceptionV3 mixed_8 + Xception ниво за редукација, (3) DenseNet121 conv5_block 16_concat + Xception ниво за редукација; и со 50% од UC Merced податочното множество за тренирање за (4) ResNet50 последно конволуциско ниво + Xception ниво за редукација, (5) Xception block14_sepconv2_act + DenseNet121 ниво за редукација

Figure 5.5 The confusion matrices of the examined method with 80% of the UC Merced dataset as a training set for (1) InceptionV3 mixed_8 + ResNet50 average pooling, (2) InceptionV3 mixed_8 + Xception average pooling, (3) DenseNet121 conv5_block 16_concat + Xception average pooling; and with 50% of the UC Merced dataset as a training set for (4) ResNet50 last conv layer + Xception average pooling, (5) Xception block14_sepconv2_act + DenseNet121 average pooling

Експериментите за предложениот метод за класификација на слики снимени од далечина со фузија на карактеристики и трансформација со анализа на главните компоненти се повторени и за WHU RS податочното множество: линеарна класификација на карактеристики споени од нивото за редукција на просек и трансформираниите карактеристики од некое од конволуциските нивоа. Точноста на класификација за различни комбинации на нивоа и класификација со линеарна машина со вектори за поддршка е прикажана во табелата 5.6. За да се оценат резултатите на испитуваната техника, ја споредуваме постигнатата вкупна точност со неколку современи методи на класификација на WHU RS податочното множество, како што е прикажано во табелата 5.7. Како што може да се забележи, испитуваната архитектура за фузија на карактеристики со намалување на димензионалноста дава точност на класификација споредлива со современите методи. Од прикажаното во табела 5.7, предложениот метод за обем на подмножеството за тренирање од 40% ги надминува сите други методи за класификација. За да се провери сигурноста на резултатите, сите случаи каде што се добива најголема точност на класификација се повторуваат десет пати. Потоа се пресметува средна вредност и стандардна девијација на постигнатите резултати, како што е прикажано во табелата 5.8.

Табела 5.6 Точност на класификација (OA (%)) за фузирани карактеристики трансформирани со анализа на главните компоненти за WHU RS
 Table 5.6 The classification accuracy (OA (%)) of fused features with PCA transformation of WHU RS dataset

Метод/ Method	60% од WHU RS/ 60% of WHU RS	40% од WHU RS/ 40% of WHU RS
ResNet50 последно конволуциско ниво (last conv layer) + InceptionV3 ниво за редукција (avg pooling)	98.26	95.02
ResNet50 последно конволуциско ниво (last conv layer) + Xception ниво за редукција (avg pooling)	97.62	96.52
DenseNet121 conv5_block16_concat + Xception ниво за редукција (avg pooling)	97.01	95.69

DenseNet121 conv4_block24_concat + Xception ниво за редукција (avg pooling)	97.76	96.68
InceptionV3 mixed_10 + ResNet50 ниво за редукција (avg pooling)	96.27	95.85
InceptionV3 mixed_8 + ResNet50 ниво за редукција (avg pooling)	98.01	98.67
InceptionV3 mixed_10 + Xception ниво за редукција (avg pooling)	96.77	96.02
InceptionV3 mixed_8 + Xception ниво за редукција (avg pooling)	98.01	96.35
DenseNet121 conv5_block16_concat + ResNet50 ниво за редукција (avg pooling)	98.76	98.34
DenseNet121 conv4_block24_concat + ResNet50 ниво за редукција (avg pooling)	96.77	96.52
Xception block14_sepconv2_act + DenseNet121 ниво за редукција (avg pooling)	97.51	96.35
Xception block14_sepconv1_act + DenseNet121 ниво за редукција (avg pooling)	97.76	96.52
DenseNet121 conv5_block16_concat + InceptionV3 ниво за редукција (avg pooling)	96.27	97.51
DenseNet121 conv4_block24_concat + InceptionV3 ниво за редукција (avg pooling)	98.01	97.18

Табела 5.7 Точност на класификација (OA (%) и SD) за методот со фузија на карактеристики и референтните методи за WHU RS

Table 5.7 The classification accuracy (OA (%) and SD) of the feature fusion method and the reference methods for WHU RS dataset

Метод/ Method	60% од WHU RS/ 60% of WHU RS	40% од WHU RS/ 40% of WHU RS
вреќа на карактеристики добиени со трансформација независно од размерот/ bag of SIFT (Chen S. et al. 2015)	85.52 ± 1.23	/

комплетни локални бинарни примероци		
во повеќе нивоа + вреќа на визулени зборови/ MS-CLBP + VoVW (Huang et al. 2016)	89.29 ± 1.30	/
GoogLeNet (Xia et al. 2017)	94.71 ± 1.33	93.12 ± 0.82
VGG-VD-16 (Xia et al. 2017)	96.05 ± 0.91	95.44 ± 0.60
CaffeNet (Xia et al. 2017)	96.24 ± 0.56	95.11 ± 1.20
salM ³ LBP-CLM (Bian et al. 2017)	96.38 ± 0.82	95.35 ± 0.76
TEX-Net-LF (Anwer et al. 2017)	96.62 ± 0.49	95.89 ± 0.37
InceptionV3 mixed_8 + ResNet50 ниво за редуција (avg pooling) (предложениот метод)	98.13 ± 0.51	/
дискриминантна анализа на корелација со собирање/ DCA by addition (Chaib et al. 2017)		
фузија со детекција на „значајност”/ fusion with saliency detection (Yu et al. 2018b)	98.70 ± 0.22	97.61 ± 0.36
фузија со детекција на „значајност”/ fusion with saliency detection (Yu et al. 2018b)	98.92 ± 0.52	98.23 ± 0.56
DenseNet121 conv5_block16_concat + ResNet50 ниво за редуција (avg pooling) (предложениот метод)	/	98.26 ± 0.40

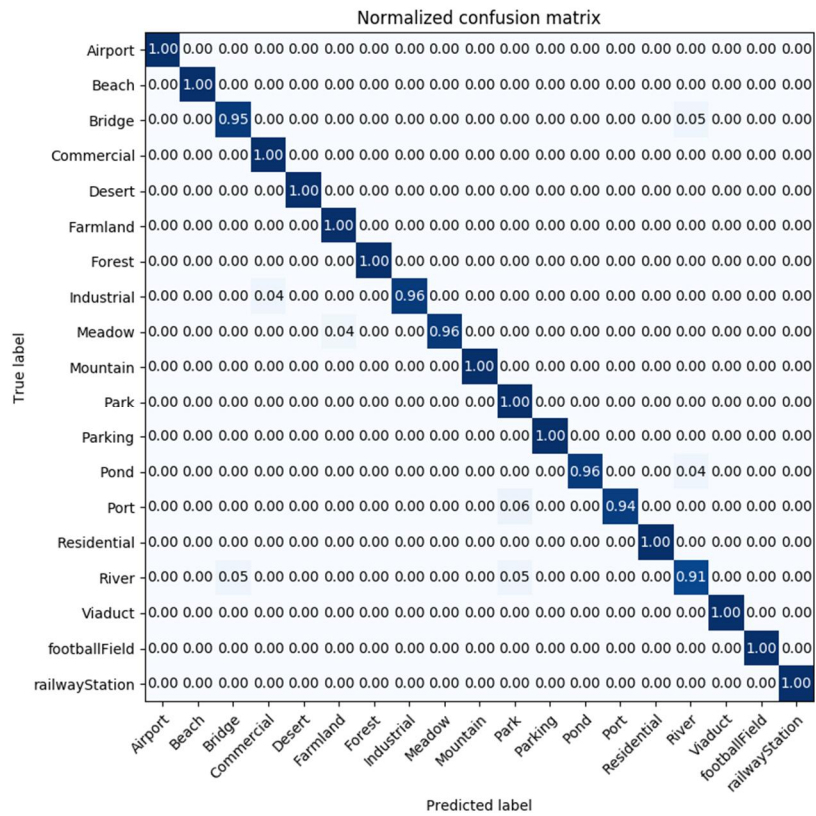
Табела 5.8 Вкупна точност (%) и стандардна девијација за предложениот метод со фузија на карактеристики за WHU RS

Table 5.8 The overall accuracy (%) and standard deviation of the proposed feature fusion method for WHU RS dataset

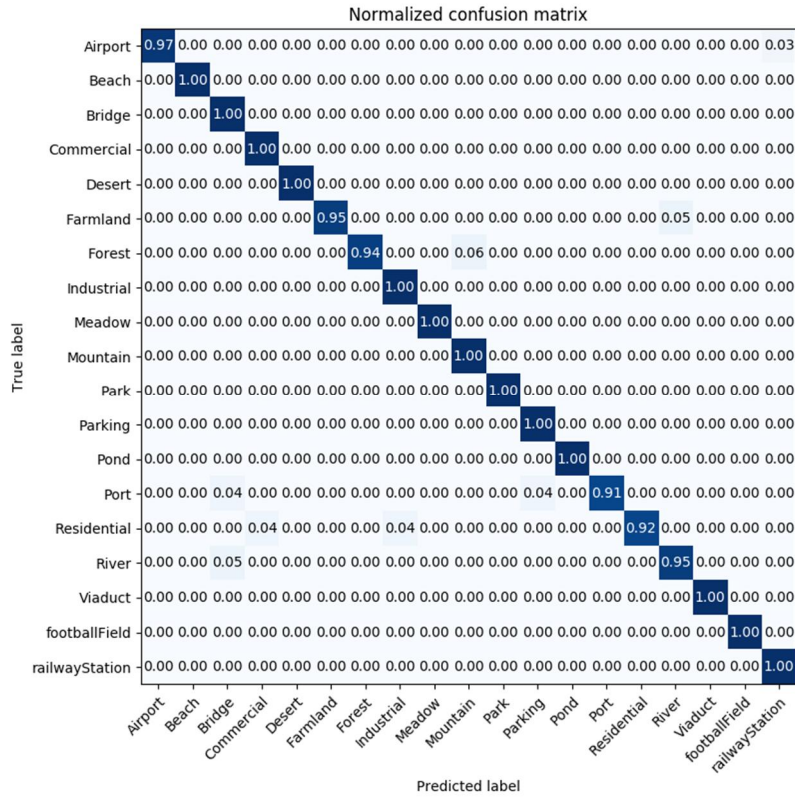
Метод/ Method	60% од WHU-RS/ 60% of WHU-RS	40% од WHU-RS/ 40% of WHU-RS
InceptionV3 mixed_8 + ResNet50 ниво за редуција (avg pooling)	98.13 ± 0.51	97.84 ± 0.53
DenseNet121 conv5_block16_concat + ResNet50 ниво за редуција (avg pooling)	98.01 ± 0.68	98.26 ± 0.40

На сликата 5.6 се дадени матриците на конфузија добиени при класификација на WHU RS податочното множество со предложениот метод на

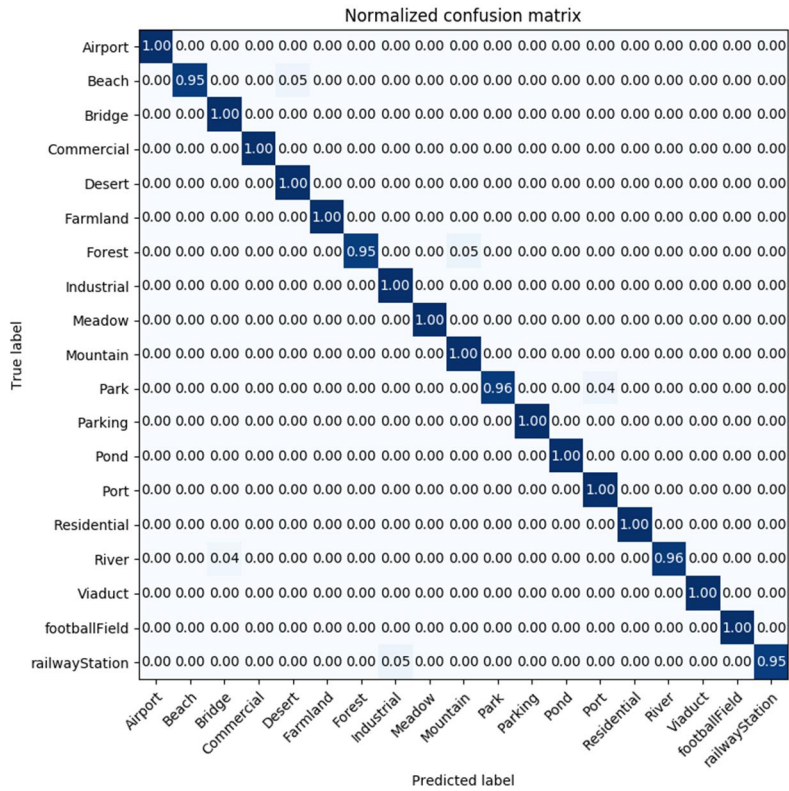
екстракција на карактеристики од различни слоеви на конволуциските мрежи и нивна фузија. Повторно се прикажани нормализирани матрици чии елементи се помали од 1.



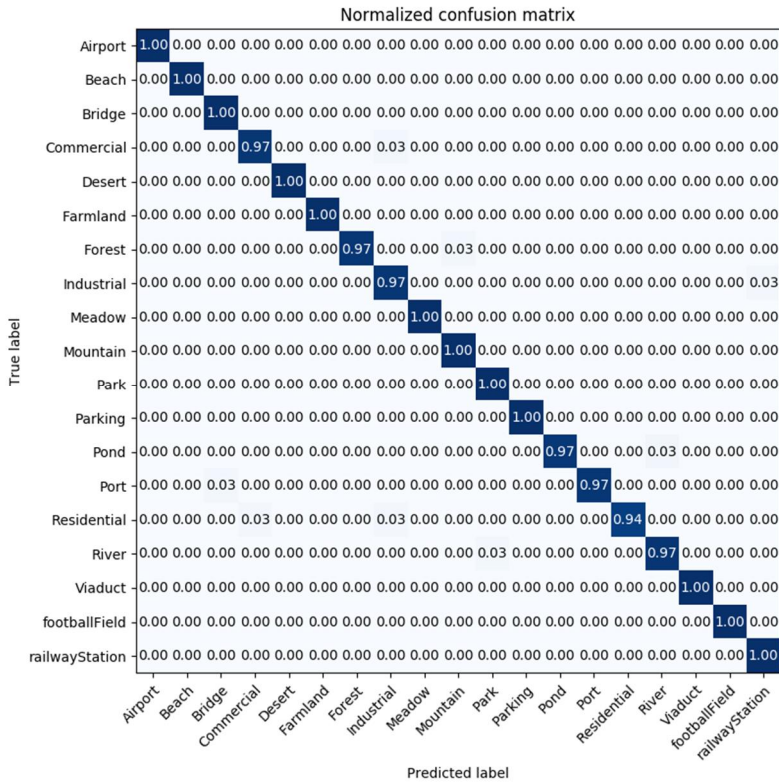
(1)



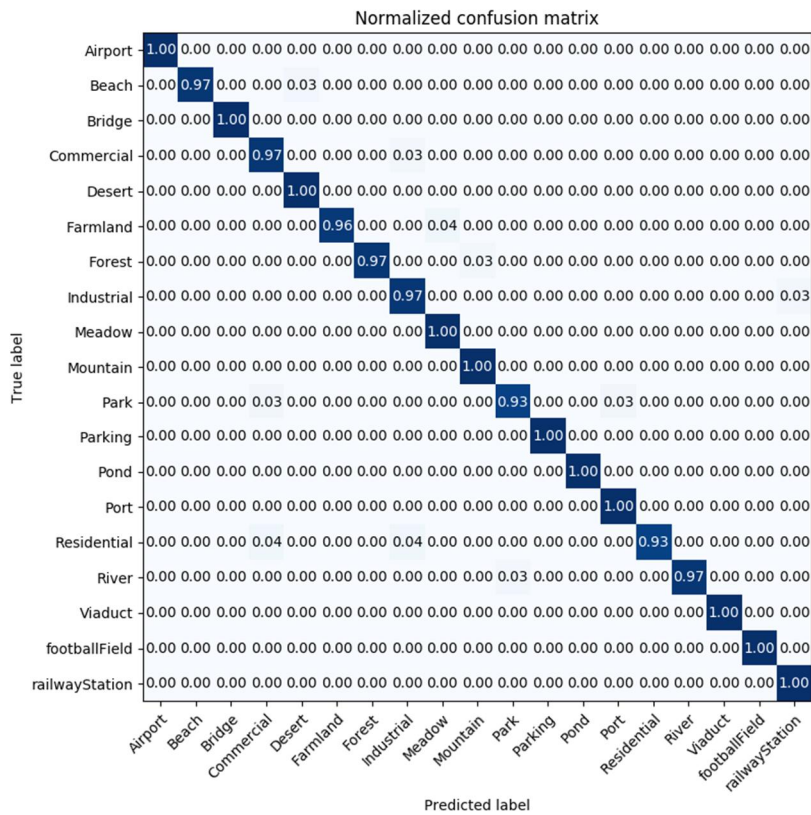
(2)



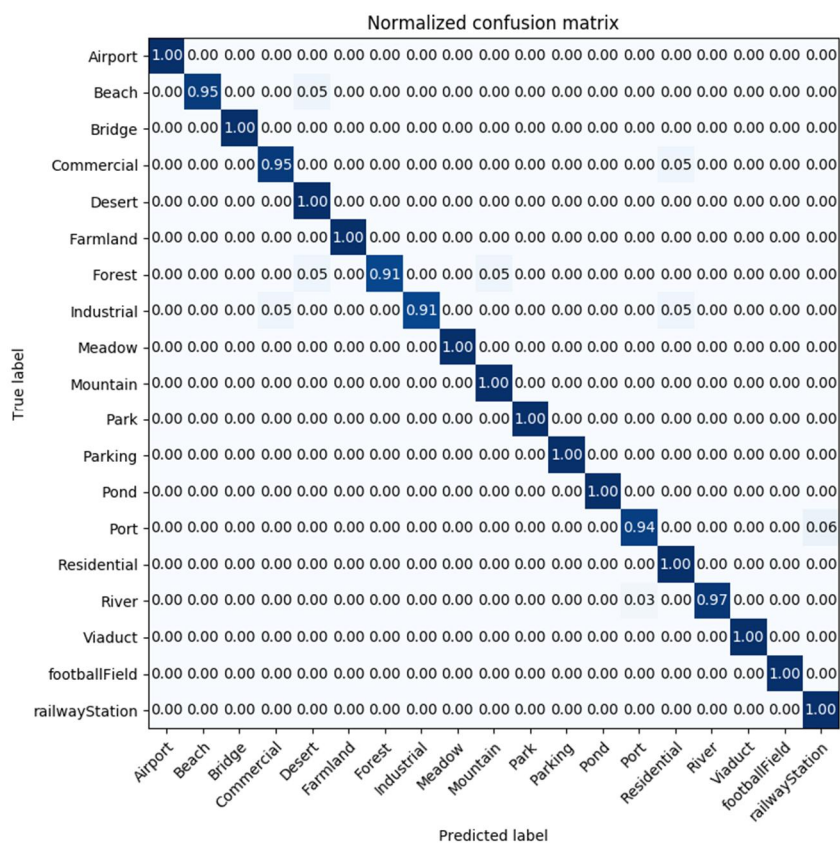
(3)



(4)



(5)



(6)

Слика 5.6 Матрици на конфузија за испитуваниот метод со 60% од WHU RS податочното множество за тренирање за (1) ResNet50 последно конволуциско ниво + InceptionV3 ниво за редукција, (2) InceptionV3 mixed_8 + ResNet50 ниво за редукција, (3) DenseNet121 conv5_block 16_concat + InceptionV3 ниво за редукција и со 40% од WHU RS податочното множество за тренирање за (4) ResNet50 последно конволуциско ниво + InceptionV3 ниво за редукција, (5) InceptionV3 mixed_8 + ResNet50 ниво за редукција, (6) DenseNet121 conv4_block 24_concat + InceptionV3 ниво за редукција

Figure 5.6 The confusion matrices of the examined method with 60% of the WHU RS dataset as a training set for 1) ResNet50 last conv layer + InceptionV3 average pooling, (2) InceptionV3 mixed_8 + ResNet50 average pooling, (3) DenseNet121 conv5_block 16_concat + InceptionV3 average pooling; and with 40% of the WHU RS dataset as a training set for (4) ResNet50 last conv layer + InceptionV3 average pooling, (5) InceptionV3 mixed_8 + ResNet50 average pooling, (6) DenseNet121 conv4_block 24_concat + InceptionV3 average pooling

5.2 Резултати од методот со фино подесување

5.2.1 Класификација на AID податочното множество

Експерименталните резултати од методот со фино подесување за класификација на AID податочното множество се прикажани во табелата 5.9 и табелата 5.10, за обем за тренирање, односно тестирање од 50%/50% и 20%/80% и класификација со линеарна машина со вектори за поддршка. Погоре споменатите размери на подмножествата за тренирање и тестирање се вообичаени во литературата и овде се користат во експериментите со цел да се спореди постигнатата точност со истражувањата од други автори. Како што може да се види од табелата 5.9, кога се користи ResNet50 и DenseNet121, чија архитектура се базира на кратки врски, линеарната машина со вектори за поддршка постигнува подобра точност на класификација споредено со софтмакс класификаторот. Меѓутоа, кога станува збор за претходно тренираните конволуциски мрежи InceptionV3 и Xception кои во својата архитектура вклучуваат „воведни“ модули, состојбата е обратна, експерименталните резултати се подобри со софтмакс класификатор споредено со класификација на екстрахираните карактеристики од фино подесените мрежи со линеарна машина со вектори за поддршка.

Табела 5.9 Вкупна точност (%) и стандардна девијација на методот со фино подесување за обем за тренирање, односно тестирање од 50%/50% за AID и линеарна машина со вектори за поддршка

Table 5.9 Overall accuracy (%) and standard deviation of the fine-tuning method with a 50%/50% train/test ratio of the AID dataset and linear SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	линеарна машина со вектори за поддршка/ linear SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	95.62±0.15	95.88±0.13
циклична стапка на учење/ cyclical learning rate	95.52±0.28	95.83±0.25
InceptionV3		

линерано опаѓање на стапката на учење/ linear decay scheduler	96.41±0.23	96.32±0.21
циклична стапка на учење/ cyclical learning rate	95.95±0.2	95.82±0.2
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	96.14±0.12	96.04±0.10
циклична стапка на учење/ cyclical learning rate	96.15±0.17	95.97±0.15
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	96.03±0.16	96.10±0.12
циклична стапка на учење/ cyclical learning rate	96.21±0.19	96.3±0.27

Табела 5.10 Вкупна точност (%) и стандардна девијација на методот со фино подесување за обем за тренирање, односно тестирање од 20%/80% за AID и линеарна машина со вектори за поддршка

Table 5.10 Overall accuracy (%) and standard deviation of the fine-tuning method with a 20%/80% train/test ratio of the AID dataset and linear SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	линеарна машина со вектори за поддршка/ linear SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.06±0.16	93.09±0.11
циклична стапка на учење/ cyclical learning rate	92.91±0.35	93.47±0.32
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.7±0.33	93.32±0.33
циклична стапка на учење/ cyclical learning rate	93.79±0.24	93.41±0.31
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.67±0.18	93.29±0.39

циклична стапка на учење/ cyclical learning rate	93.44±0.10	93.36±0.18
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.74±0.24	93.26±0.29
циклична стапка на учење/ cyclical learning rate	93.54±0.15	93.35±0.22

Анализата на табела 5.10, која ги дава експерименталните резултати за обем на подмножествата за тренирање, односно тестирање од 20%/80%, покажува малку поинаков исход: софтмакс класификаторот работи подобро за InceptionV3, Xception, и DenseNet121, но класификацијата со линеарна машина со вектори за поддршка е подбра опција за ResNet50.

Во втората фаза на истражувањето на методот на пренос на учење со фино подесување за класификација на податочни множества од воздушни сцени се користи машина со вектори за поддршка со Гаусово (RBF) јадро. За разлика од истражувањето во првата фаза, каде што експериментите за секое симулационо сценарио се повторуваат по пет пати, а потоа се пресметува средна вредност и стандардна девијација, овде сите експерименти се извршени само еднаш. Целта на вака изведените симулациони сценарија е да се испита можноста од понатамошно зголемување на точност на класификација, која и така е веќе респектабилно висока споредено со современите техники. Табелите 5.11 и 5.12 ја прикажуваат постигнатата точност на класификација на методот со фино подесување за класификација на AID податочното множество за обем за тренирање, односно тестирање од 50%/50% и 20%/80% и класификација со машина со вектори за поддршка со нелинеарно (Гаусово) јадро.

Табела 5.11 Вкупна точност (%) на методот со фино подесување за обем за тренирање/ тестирање од 50%/50% за AID и машина со вектори за поддршка со Гаусово јадро

Table 5.11 Overall accuracy (%) of the fine-tuning method with a 50%/50% train/test ratio of the AID dataset and RBF SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	машина со вектори за поддршка со Гаусово
---------------	---	---

		јадро / RBF SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	95.66	96.12
циклична стапка на учење/ cyclical learning rate	95.64	96.08
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	95.88	95.96
циклична стапка на учење/ cyclical learning rate	96.00	96.18
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	95.96	95.96
циклична стапка на учење/ cyclical learning rate	96.28	96.3
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	96.44	98.03
циклична стапка на учење/ cyclical learning rate	96.3	96.6

Споредбата на софтверскиот класификатор и класификацијата со нелинеарна машина со вектори за поддршка на AID податочното множество покажува дека нелинеарната машина го надминува софтверскиот класификаторот за обемот за тренирање, односно тестирање од 50%/50% во сите симулациони сценарија, освен за Xception невронската мрежа со линеарно опаѓање на стапката за учење. За обемот за тренирање, односно тестирање од 20%/80% за AID податочното множество, нелинеарната машина со вектори за поддршка постигнува подобра точност на класификација од софтверскиот класификатор во сите експериментални сценарија.

Табела 5.12 Вкупна точност (%) на методот со фино подесување за обем за тренирање, односно тестирање од 20%/80% за AID и машина со вектори за поддршка со Гаусово јадро

Table 5.12 Overall accuracy (%) of the fine-tuning method with a 20%/80% train/test ratio of the AID dataset and RBF SVM

Метод/ Method	софтмакс	машина со вектори за
	класификатор/ softmax classifier	поддршка со Гаусово јадро/ RBF SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	91.99	92.98
циклична стапка на учење/ cyclical learning rate	93.24	93.44
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.33	93.5
циклична стапка на учење/ cyclical learning rate	93.81	93.93
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.89	94.14
циклична стапка на учење/ cyclical learning rate	93.49	93.65
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.28	93.56
циклична стапка на учење/ cyclical learning rate	93.25	93.58

Во табелата 5.13 е дадена споредба на предложениот метод со фино подесување со други современи методи. Важно е да се напомене дека при споредбата се земени во предвид резултатите кои се добиени со линеарна машина со вектори за поддршка, бидејќи се повторени по пет пати и се сметаат за сигурни. Најдобри резултати за класификација на AID податочното множество со обем за тренирање од 50% се добиваат за InceptionV3 со линеарно опаѓање на стапката на учење и софтмакс класификатор, а за обем за тренирање од 20%

за InceptionV3 со циклична стапка на учење и софтмакс класификатор. Ако се потсетиме на резултатите прикажани во табелите 5.11 и 5.12, каде што машината со вектори за поддршка со Гаусово јадро го надминува софтмакс нивото во сите случаи, тогаш во табелата 5.13 би можеле и да напредуваме неколку места погоре. Меѓутоа, прикажаните резултати и вака се респектабилно високи. Постојат методи кои даваат подобри резултати при класификација споредено со предложениот метод, како EfficientNet-V3-aux (Bazi et al. 2019). Во посочениот труд авторите користат fino подесување на EfficientNet-V3 мрежата со помошен класификатор. Причината за подобро достигнување можеби лежи во фактот дека претходно тренираната мрежа и самата има подобра топ-1 точност на класификација од конволуциските мрежи кои овде се користат. Други автори кои објавиле подобра точност на класификација од овде се (Zeng et al. 2018) и (Zhang W. et al 2019, Zhang J. et al 2019), каде што експериментите се базирани на фузија на повеќе карактеристики, кои се извлекуваат од сликите или од нивни делови со различни димензии (размери). Овде се користи fino подесување со една претходно дефинирана големина на сликите одредена од мрежната архитектура.

Табела 5.13 Вкупна точност (%) и стандардна девијација на методот со fino подесување споредена со референтни методи за обем од 50% и 20% од AID за тренирање

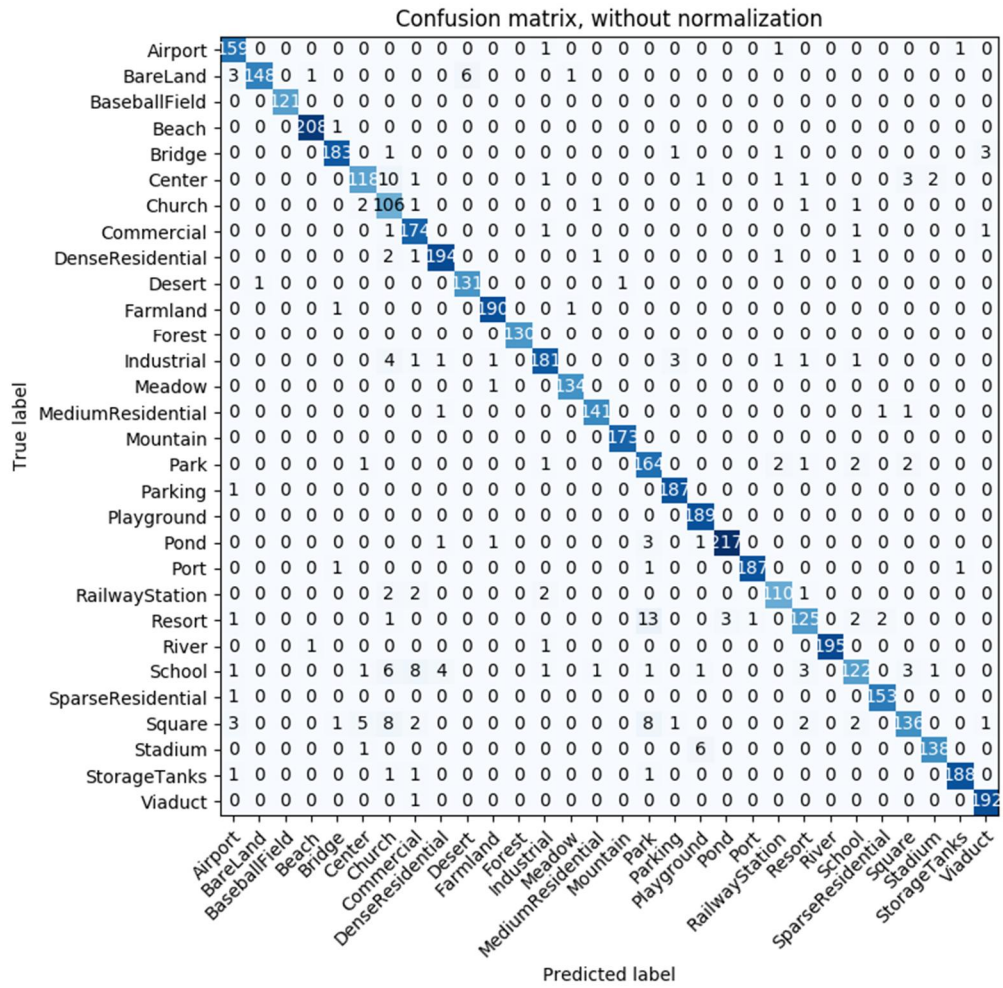
Table 5.13 Overall accuracy (%) and standard deviation of the fine-tuning method compared to reference methods with 50% and 20% of the AID dataset as training set

Метод/ Method	50% слики за тренирање/ 50% training ratio	20% слики за тренирање/ 20% training ratio
GoogLeNet+SVM (Xia et al. 2017)	86.39±0.55	83.44±0.40
VGG-VD-16 (Xia et al. 2017)	89.64±0.36	86.59±0.29
CaffeNet (Xia et al. 2017)	89.53±0.31	86.86±0.46
salM ³ LBP-CLM (Bian et al. 2017)	89.76±0.45	86.92±0.35
конволуциска мрежа во повеќе нивоа/ MCNN (Liu et al 2018a)	91.80±0.22	/
фузија со додавање/ fusion by addition (Wang et al. 2017)	91.87±0.36	/
TEX-Net-LF (Anwer et al. 2017)	92.96±0.18	90.87±0.11

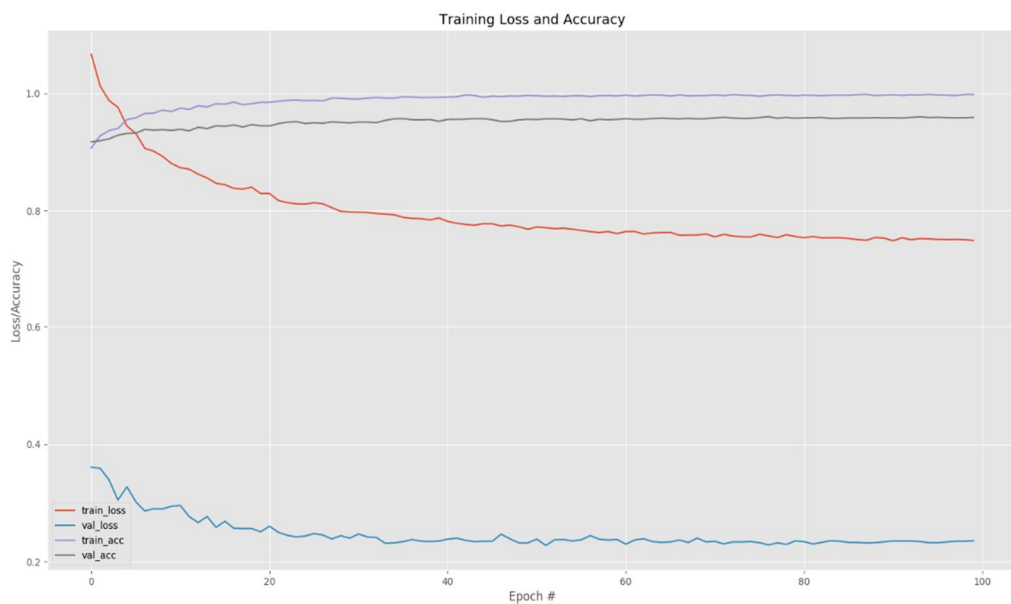
рекурентна конволуциска мрежа на привлекување внимание/ ARCNet-VGG16 (Wang et al. 2018)	93.10±0.55	88.75±0.40
фино подесување на VGG-16/ VGG-16 fine-tuning (Sun et al. 2019)	93.60±0.64	89.49±0.34
VGG-16+MSCP (He et al. 2018)	94.42±0.17	91.52±0.21
фузија од два извора/ two-stream fusion (Yu et al. 2018b)	94.58±0.25	92.32±0.41
фузија од повеќе извори/ multilevel fusion (Yu et al. 2018c)	95.36±0.22	/
GBNet + глобални карактеристики/ GBNet + global Feature (Sun et al. 2019)	95.48±0.25	92.20±0.23
InceptionV3-CapsNet (Zhang W. et al. 2019)	96.32±0.12	93.79±0.13
InceptionV3 со линеарно опаѓање на стапката на учење и софтмакс класификатор (предложениот метод)/ InceptionV3 with linear decay scheduler and softmax classifier	96.41±0.23	93.7±0.33
InceptionV3 со циклична стапка на учење и софтмакс класификатор (предложениот метод)/ InceptionV3 with cyclical learning rate and softmax classifier	95.95±0.2	93.79±0.24
EfficientNet-B3-aux (Bazi et al.2019)	96.56±0.14	94.19±0.15
карактеристики со глобален контекст + карактеристики кои се однесуваат на локалните објекти/ GCFs + LOFs (Zeng et al. 2018)	96.85±0.23	92.48±0.38
D-CNN + VGGNet-16 (Cheng et al. 2018)	96.89±0.1	90.82±0.16
„густа“ конволуциска мрежа + 3-D редукција/ Dense-based CNN + 3D pooling (Zhang J. et al. 2019)	97.19±0.23	95.37±0.32

Сликата 5.7 прикажува матрици на конфузија и графици на тренирање за AID податочното множество. Сите матрици на конфузија се дадени без нормализација, односно нивните членови ја даваат бројната вредност на сликите од различни категории кои се точно или погрешно (мрежата ги „измешала“ со слики од други класи) класифицирани. Дадени се по три повторувања за фино подесување на ResNet50 со линеарно опаѓање на стапката на учење и со циклични стапки на учење и класификација со софтмакс нивото за обем за тренирање, односно тестирање од 50%/50%. Понатаму прикажани се и три повторувања за фино подесување на ResNet50 со линеарно

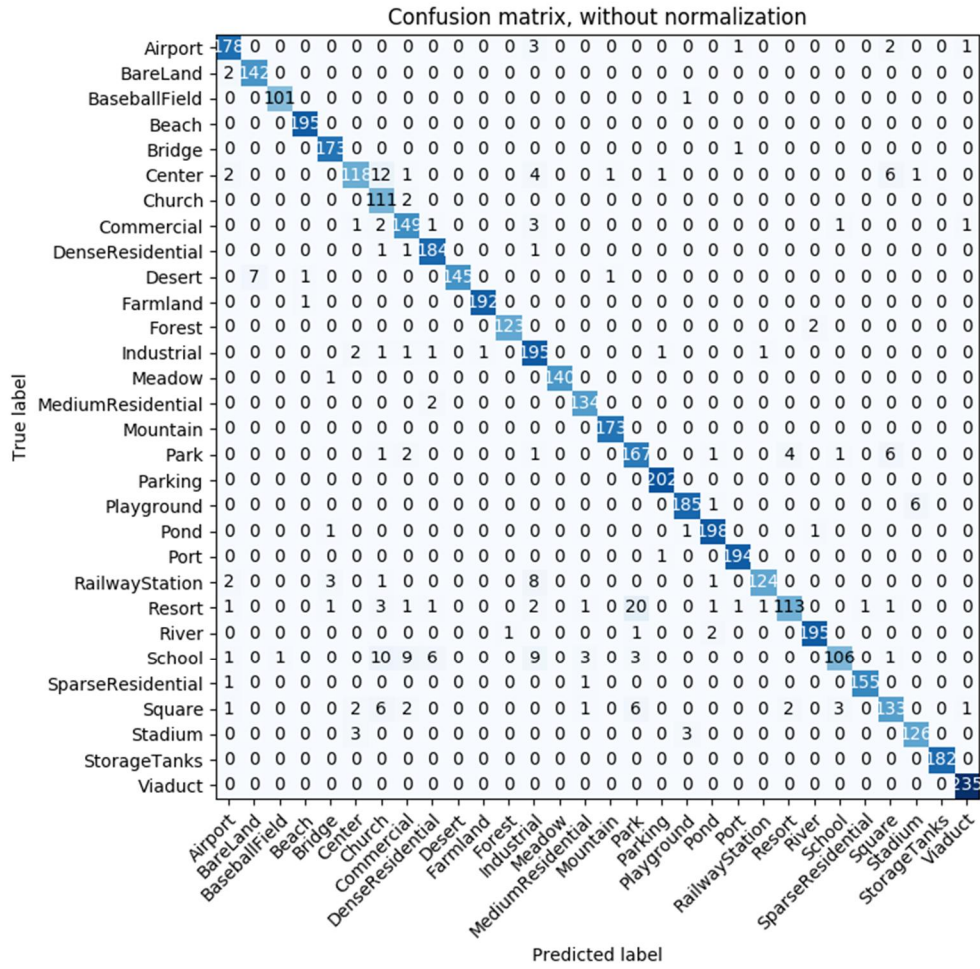
опаѓање на стапката на учење и софтвакс класификатор за обем за тренирање/
тестирање од 20%/80%.



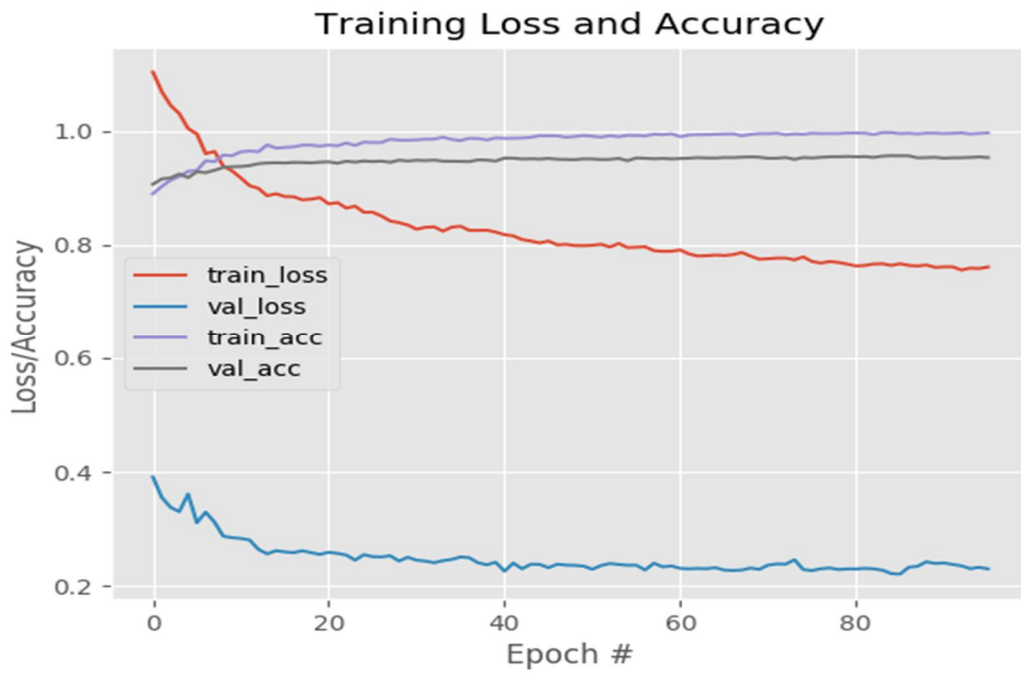
(1a)



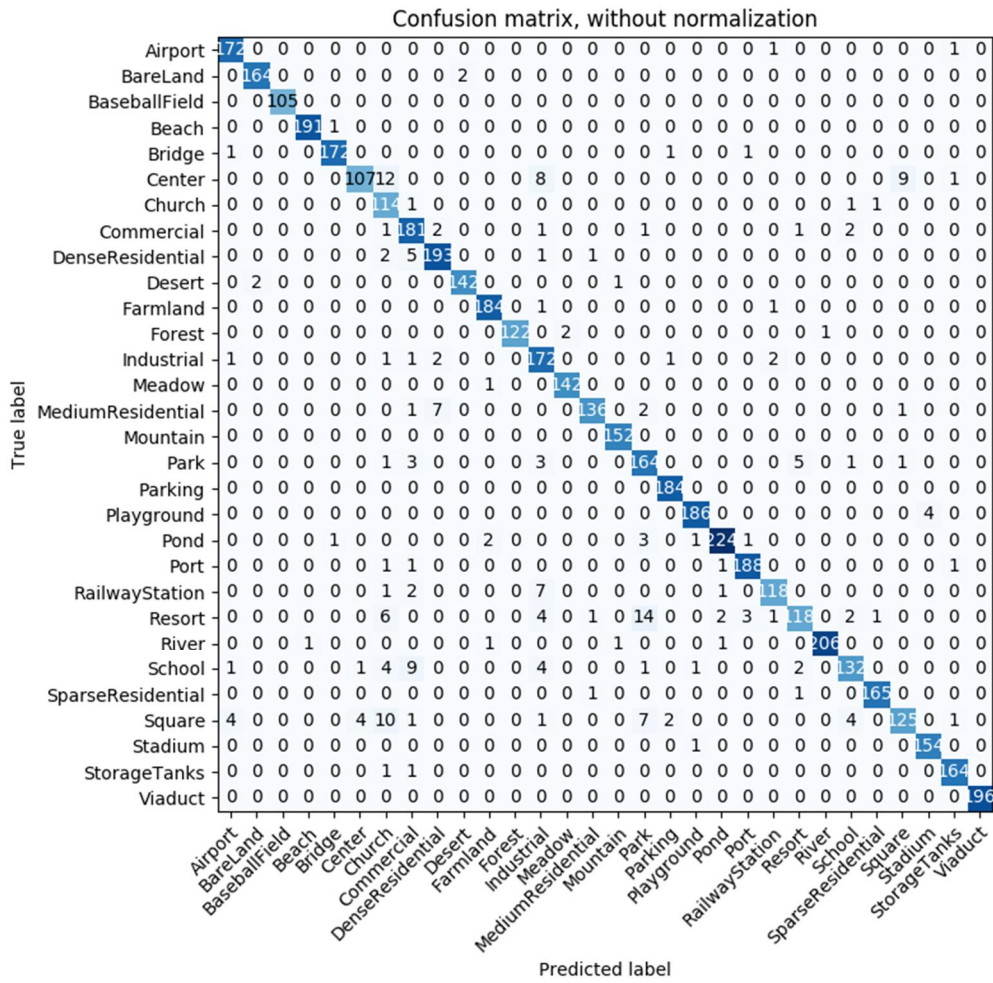
(16)



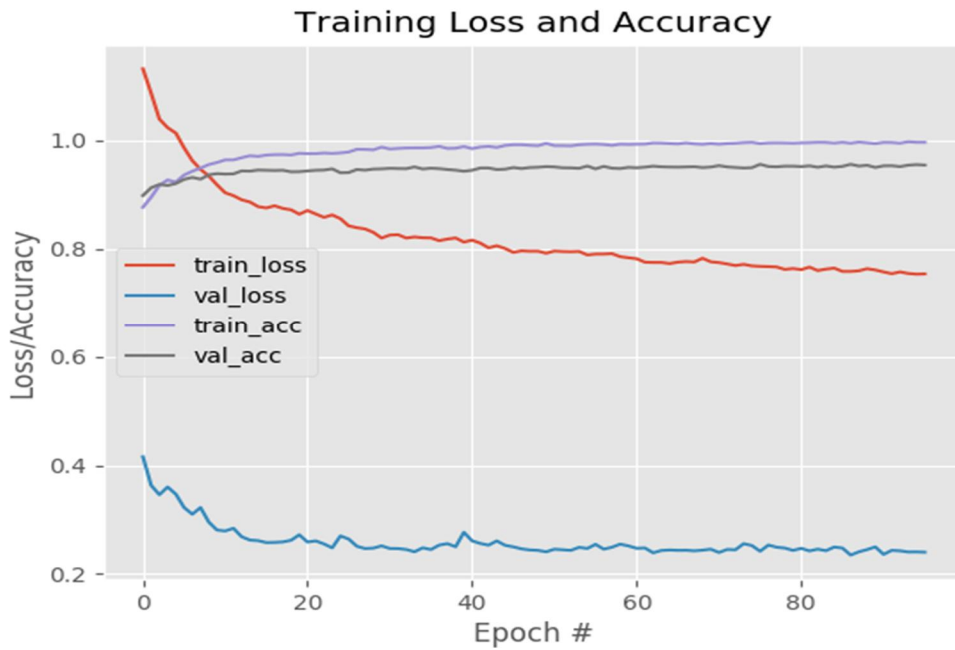
(4a)



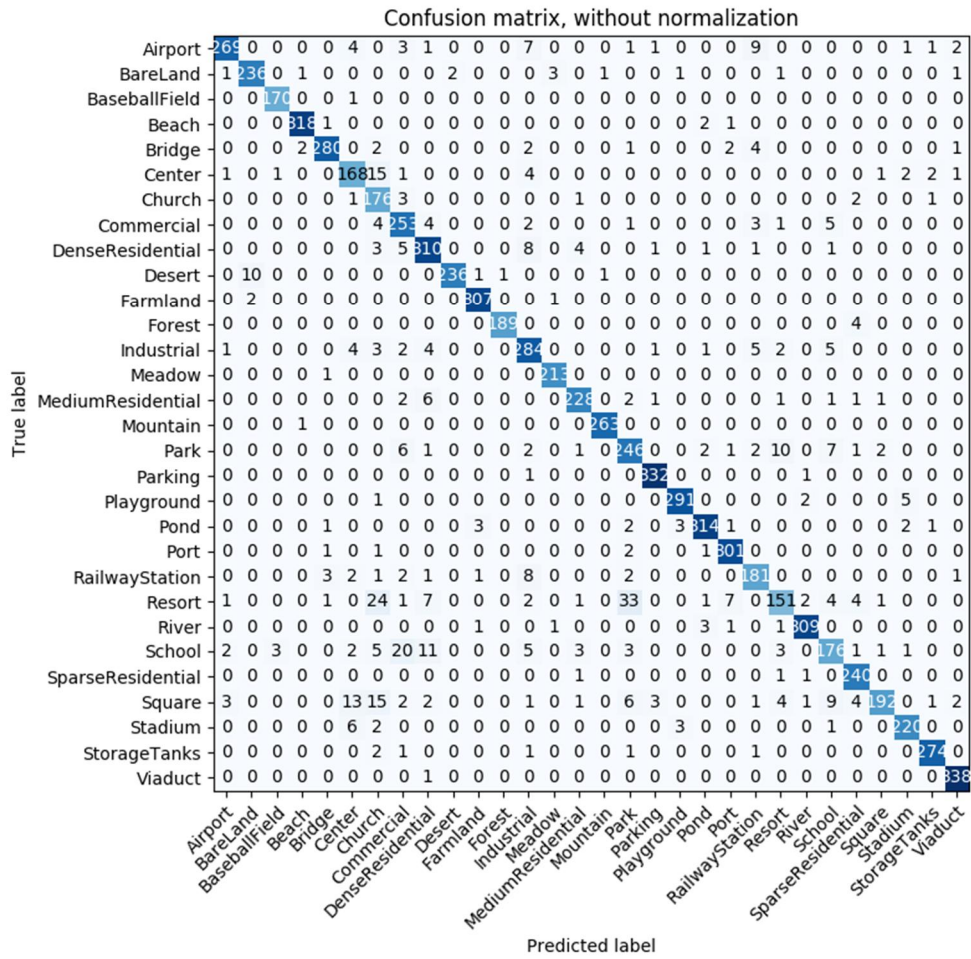
(46)



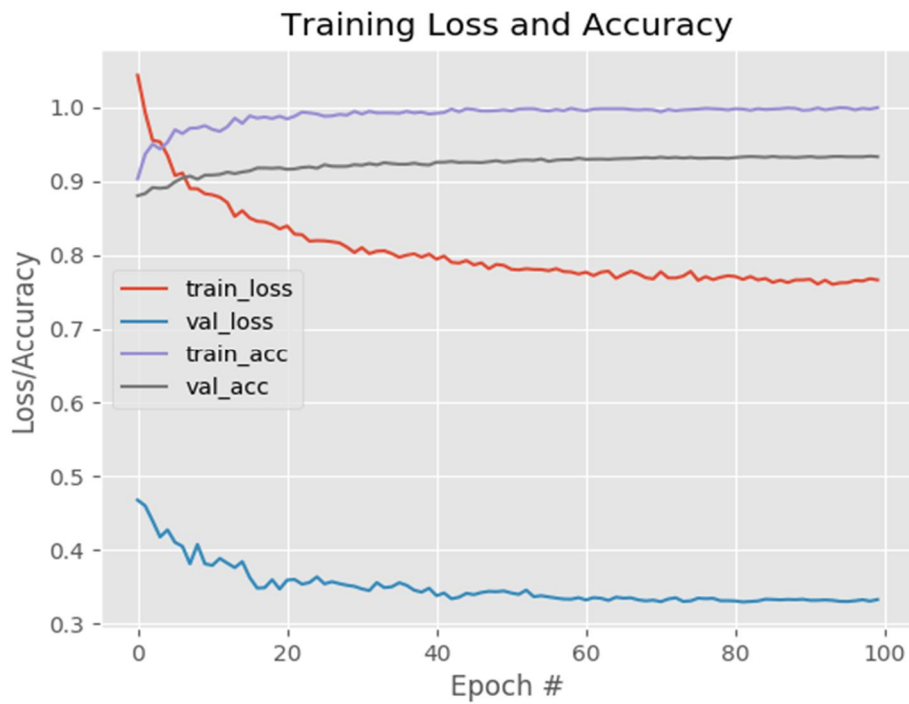
(6a)



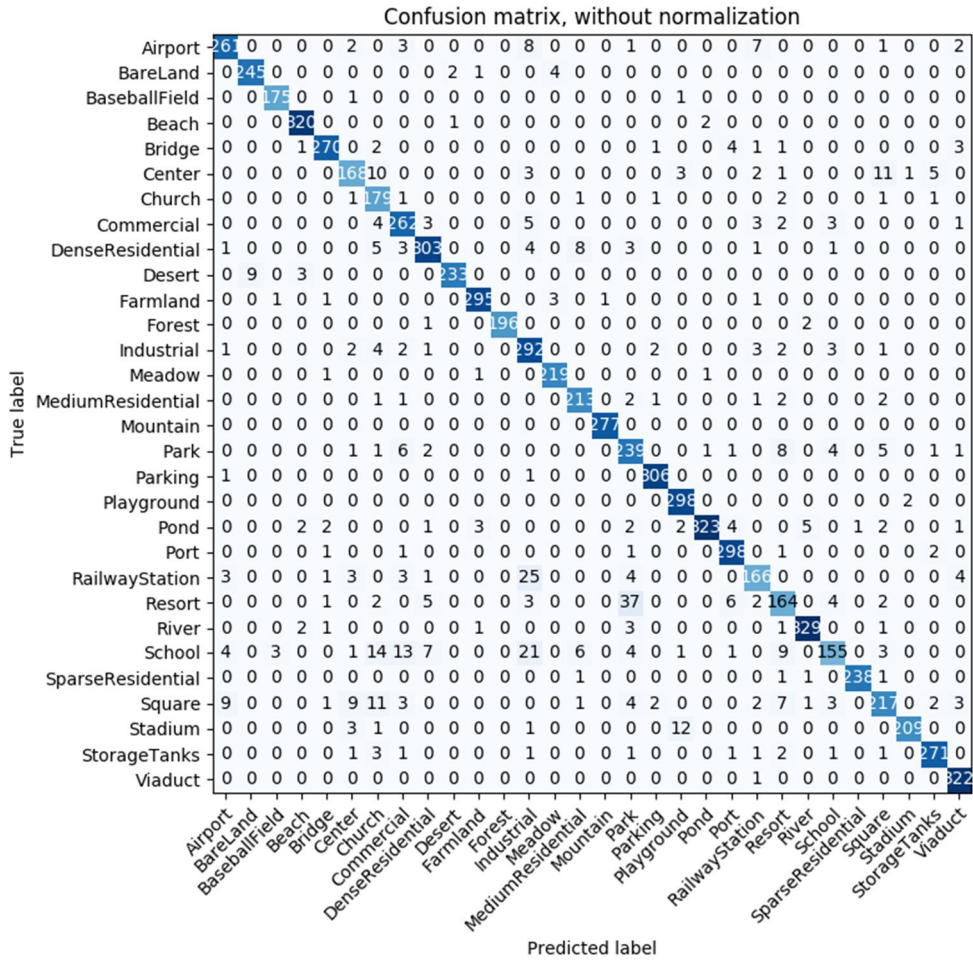
(66)



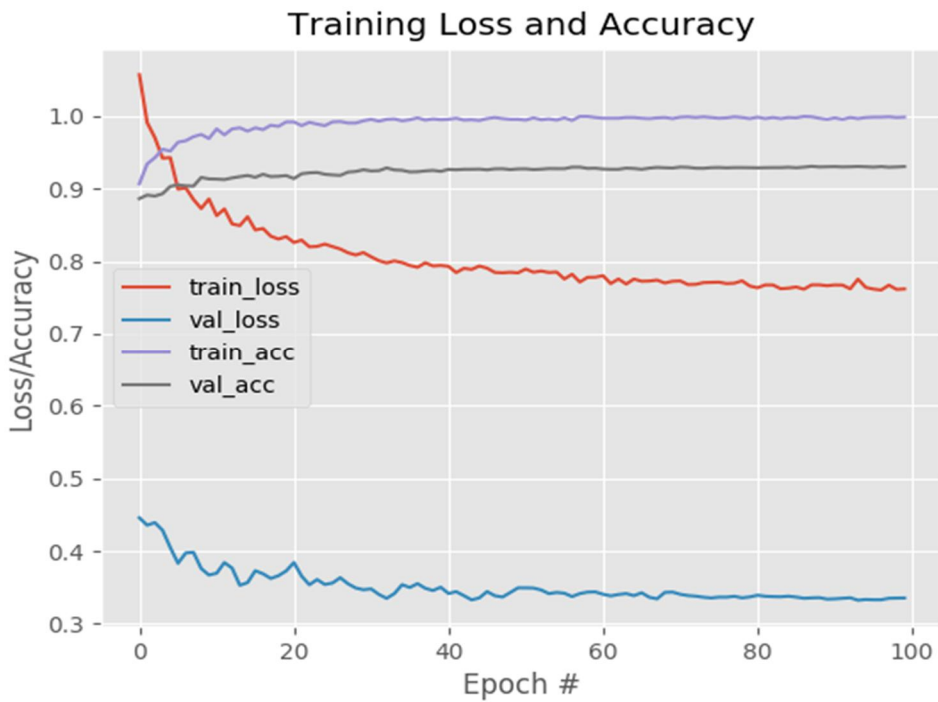
(7a)



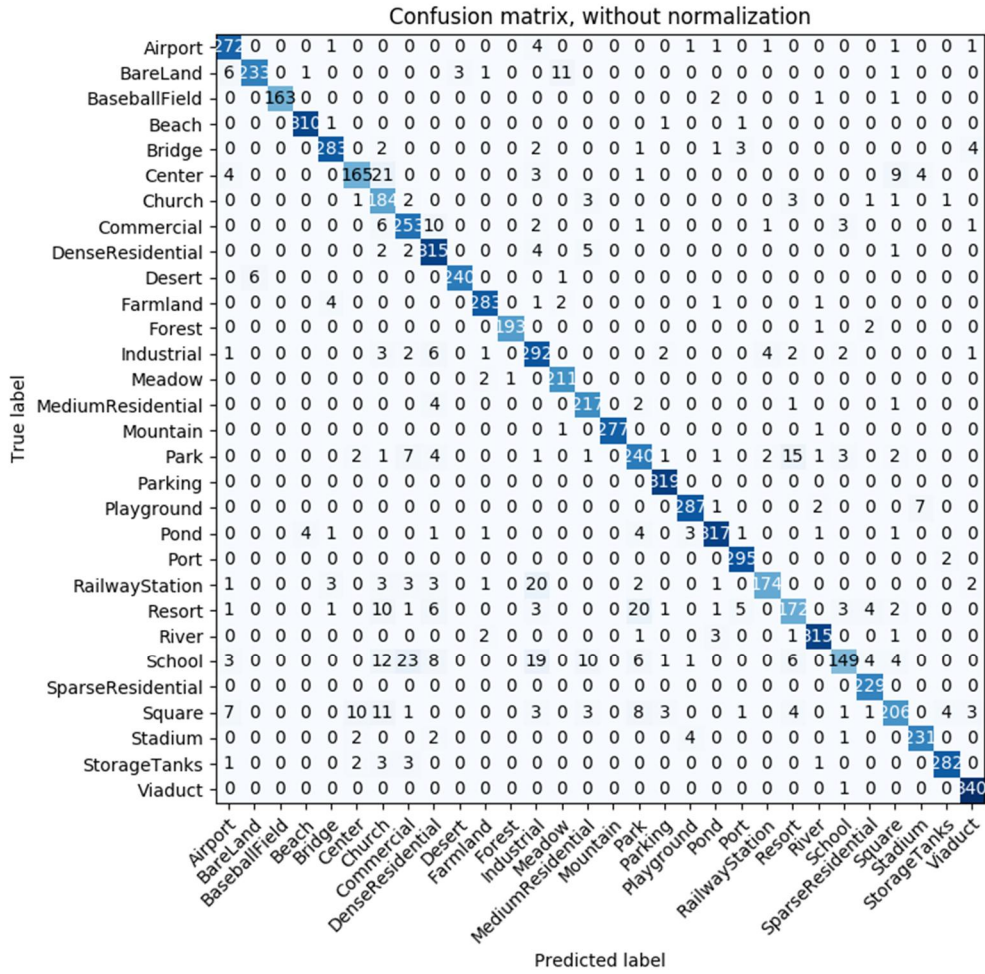
(76)



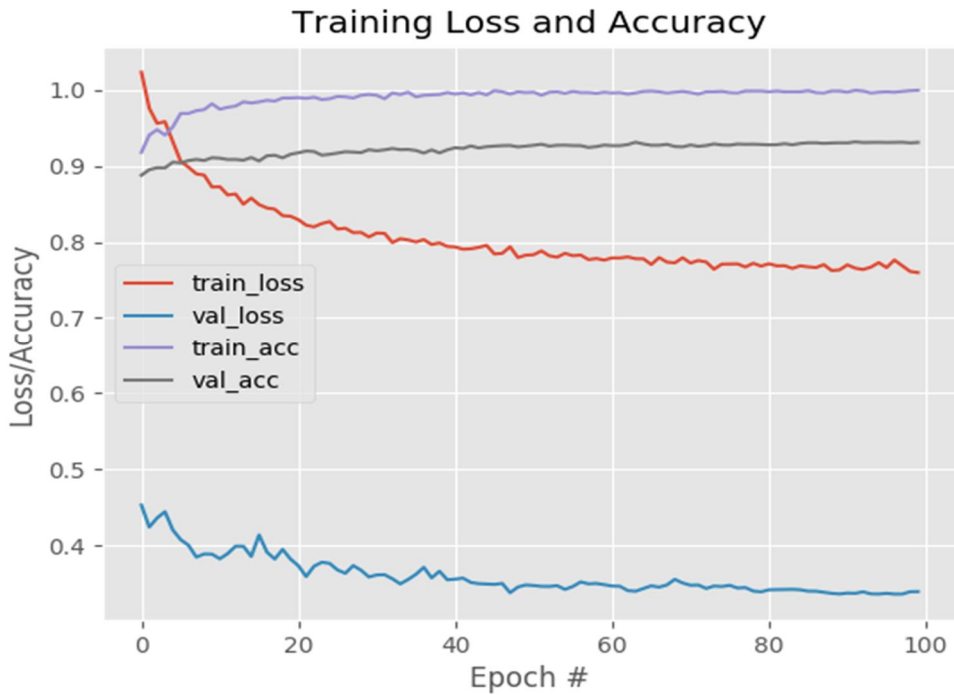
(8a)



(8b)



(9a)



(96)

Бидејќи точноста на класификација со софтмакс нивото и линеарната машина е блиска, овие матрици се разликуваат само во класификацијата на мал број слики. Графиците на фино подесување го даваат тренирањето на сите мрежни нивоа со стохастички градиентен спуст, но не и загревањето на новата мрежна глава. Од овие графици може да се види дека тренирањето со циклични стапки на учење е постабилно со карактеристични врвови на кривите на загуба при тренирање и валидација (тестирање), создавајќи одредени осцилации кои се видливи како брановидна форма. Исто така, се забележува дека загубите при тренирање се поголеми од загубите при валидација (тестирање) бидејќи регуларизационата техника измазнување на лабели се применува само на подмножеството за тренирање.

5.2.2 Класификација на NWPU-RESISC45 податочното множество

Експерименталните резултати од методот со фино подесување и линеарната машина со вектори за поддршка за NWPU-RESISC45 податочното множество се дадени во табелите 5.14 и 5.15 и сликата 5.8. Табелата 5.14 ја покажува точноста на класификација за обем за тренирање, односно тестирање од 20%/80% од податочното множество. Може да се забележи дека за линеарно опаѓање на стапката на учење како и за циклични стапки на учење, класификацијата со линеарна машина со вектори за поддршка дава подобра вкупна точност споредено со софтмакс класификатор за сите претходно тренирани конволуциски невронски мрежи. Табелата 5.15 ја покажува постигнатата точност на класификација за обем на тренирање, односно тестирање од 10%/90% од NWPU-RESISC45 податочното множество. Тука експерименталните резултати се слични со оние од табелата 5.14. Линеарната машина со вектори за поддршка го надминува софтмакс класификаторот во сите случаи, освен при фино подесување на InceptionV3 мрежата за двата вида стапки на учење и Xception мрежата со линеарно опаѓање на стапката на учење. Двата односа на обемот за тренирање, односно тестирање од анализираното податочно множество се избрани со цел да се направи споредба со другите истражувања од областа, кои ги користат истите односи на поделба.

Табела 5.14 Вкупна точност (%) и стандардна девијација на методот со фино подесување за обем за тренирање, односно тестирање од 20%/80% за NWPU-RESISC45 и линеарна машина со вектори за поддршка

Table 5.14 Overall accuracy (%) and standard deviation of the fine-tuning method with a 20%/80% train/test ratio of the NWPU-RESISC45 dataset and linear SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	линеарна машина со вектори за поддршка/ linear SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	92.35±0.10	92.77±0.13
циклична стапка на учење/ cyclical learning rate	92.40±0.10	92.85±0.12
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.07±0.17	93.18±0.10
циклична стапка на учење/ cyclical learning rate	93.04±0.16	93.13±0.16
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	92.63±0.20	92.78±0.17
циклична стапка на учење/ cyclical learning rate	92.63±0.28	92.80±0.29
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.16±0.23	93.37±0.12
циклична стапка на учење/ cyclical learning rate	92.98±0.25	93.26±0.25

Табела 5.15 Вкупна точност (%) и стандардна девијација на методот со фино подесување за обем за тренирање, односно тестирање од 10%/90% за NWPU-RESISC45 и линеарна машина со вектори за поддршка

Table 5.15 Overall accuracy (%) and standard deviation of the fine-tuning method with a 10%/90% train/test ratio of the NWPU-RESISC45 dataset and linear SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	линеарна машина со вектори за поддршка/ linear SVM classifier
ResNet50		

линерано опаѓање на стапката на учење/ linear decay scheduler	89.42±0.38	89.74±0.27
циклична стапка на учење/ cyclical learning rate	89.20±0.12	89.70±0.17
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	90.16±0.19	90.07±0.26
циклична стапка на учење/ cyclical learning rate	90.21±0.10	90.18±0.13
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	89.62±0.13	89.59±0.07
циклична стапка на учење/ cyclical learning rate	89.40±0.16	89.65±0.20
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	90.25±0.20	90.46±0.22
циклична стапка на учење/ cyclical learning rate	89.73±0.25	89.99±0.12

Втората фаза на истражувањето на методот на пренос на учење со фино подесување со употреба на нелинеарна машина со вектори за поддршка е повторена и за класификација на NWPU-RESISC45 податочното множество. Експериментите за машината со вектори за поддршка со Гаусово јадро се спроведуваат само еднаш. Иако точноста на класификација на екстрахираните карактеристики од фино подесените конволуциски мрежи со линеарна машина ја надминува онаа добиена со софтмакс нивото во најголем дел од случаите, сепак се прави обид за дополнително подобрување на експерименталните резултати со воведување на машина со вектори за поддршка со нелинеарно јадро. Табелата 5.16 и табелата 5.17 ја прикажуваат постигнатата точност на класификација со методот со фино подесување за класификација на NWPU-RESISC45 податочното множество за обем за тренирање, односно тестирање од 20%/80% и 10%/90% и класификација со машина со вектори за поддршка со Гаусово јадро.

Табела 5.16 Вкупна точност (%) на методот со фино подесување за обем за тренирање, односно тестирање од 20%/80% за NWPU-RESISC45 и машина со вектори за поддршка со Гаусово јадро

Table 5.16 Overall accuracy (%) of the fine-tuning method with a 20%/80% train/test ratio of the NWPU-RESISC45 dataset and RBF SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	машина со вектори за поддршка со Гаусово јадро/ RBF SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	92.71	92.89
циклична стапка на учење/ cyclical learning rate	92.43	92.77
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	93.17	93.35
циклична стапка на учење/ cyclical learning rate	92.52	92.82
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	92.36	92.72
циклична стапка на учење/ cyclical learning rate	92.72	92.87
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	92.98	93.60
циклична стапка на учење/ cyclical learning rate	93.11	93.55

Споредбата на софтмакс и класификацијата со нелинеарна машина со вектори за поддршка на NWPU-RESISC45 податочното множество покажува дека нелинеарната машина го надминува софтмакс класификаторот за обемот за тренирање, односно тестирање од 20%/80% во сите симулациони сценарија. За обемот за тренирање, односно тестирање од 10%/90% за NWPU-RESISC45 податочното множество, машината со Гаусово јадро постигнува подобра точност на класификација од софтмакс нивото во сите експериментални сценарија,

освен за Xception невронската мрежа со линеарно опаѓање на стапката за учење.

Табела 5.17 Вкупна точност (%) на методот со фино подесување за обем за тренирање, односно тестирање од 10%/90% за NWPU-RESISC45 и машина со вектори за поддршка со Гаусово јадро

Table 5.17 Overall accuracy (%) of the fine-tuning method with a 10%/90% train/test ratio of the NWPU-RESISC45 dataset and RBF SVM

Метод/ Method	софтмакс класификатор/ softmax classifier	машина со вектори за поддршка со Гаусово јадро/ RBF SVM classifier
ResNet50		
линерано опаѓање на стапката на учење/ linear decay scheduler	89.45	90.03
циклична стапка на учење/ cyclical learning rate	89.35	89.99
InceptionV3		
линерано опаѓање на стапката на учење/ linear decay scheduler	90.16	90.36
циклична стапка на учење/ cyclical learning rate	90.10	90.36
Xception		
линерано опаѓање на стапката на учење/ linear decay scheduler	89.27	89.17
циклична стапка на учење/ cyclical learning rate	89.25	89.67
DenseNet121		
линерано опаѓање на стапката на учење/ linear decay scheduler	89.99	90.42
циклична стапка на учење/ cyclical learning rate	89.51	90.11

Во табелата 5.18 е дадена споредба на испитуваната техника со фино подесување и други современи методи. И овде, како и при класификација на AID податочното множество, предвид се земени само резултатите кои се добиени со линеарна машина со вектори за поддршка, бидејќи се повторени по пет пати и

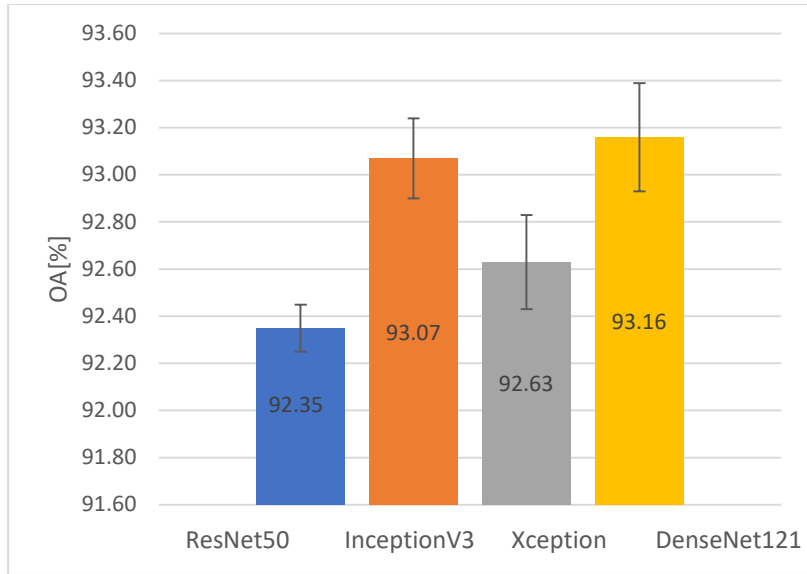
се сметаат за веродостојни. Предложената техника постигнува најдобра точност на класификација со DenseNet121 со линеарно опаѓање на стапката на учење и линеарна машина за обем на тренирање, односно тестирање од 20%/80% и од 10%/90% за NWPU-RESISC45 податочното множество. Ако ги разгледаме резултатите од табелите 5.16 и 5.17, може да се заклучи дека со класификација со машина со вектори за поддршка со Гаусово јадро се добиваат подобри експериментални резултати во однос на линеарната машина со вектори за поддршка. Тоа значи дека во табелата 5.18 би можеле да прикажеме и подобри резултати, но и вака точноста на класификација е задоволителна. Од табелата 5.18 може да се заклучи дека постојат методи кои го надминуваат предложениот метод. Еден од нив користи фино подесување на EfficientNet-B3 мрежата со помошен класификатор (Bazi et al. 2019). EfficientNet-B3 дава подобра топ-1 и топ-5 точност на класификација на ImageNet податочното множество споредено со претходно тренираните конволуциски мрежи кои се користат во истражувањата во овој докторски труд, а тоа е веројатно главната причина за подобрата постигната точност. Zhang J. et al (2019) исто така прикажуваат подобри резултати од добиените овде, но тие користат повеќекратна фузија на карактеристики екстрахирани од сликите од множеството или од нивни делови со различни димензии (размери). Наспроти ова, тука се користи фино подесување со една големина на сликите согласно со барањата на претходно тренираните конволуциски мрежи.

Табела 5.18 Вкупна точност (%) и стандардна девијација на методот со фино подесување споредена со референтни методи за обем од 20% и 10% од NWPU-RESISC45 за тренирање

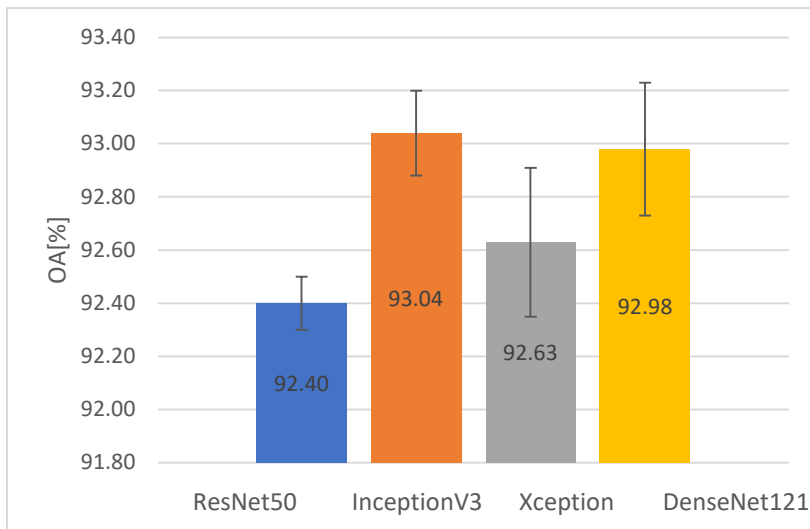
Table 5.18 Overall accuracy (%) and standard deviation of the fine-tuning method compared to reference methods with 20% and 10% of the NWPU-RESISC45 dataset as a training set

Метод/ Method	20% слики за тренирање/ training ratio	10% слики за тренирање/ training ratio
GoogLeNet (Cheng et al. 2017a)	78.48±0.26	76.19±0.38
VGG-16 (Cheng et al. 2017a)	79.79±0.15	76.47±0.18
AlexNet (Cheng et al. 2017a)	79.85±0.13	76.69±0.21

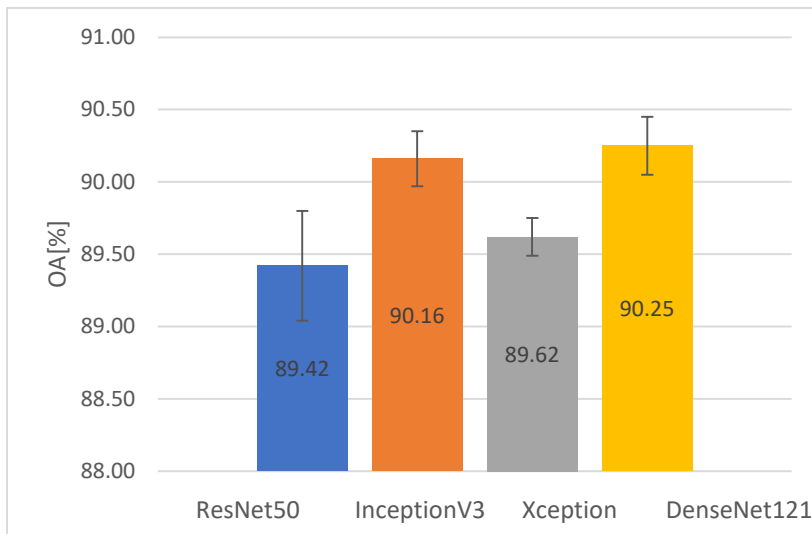
фузија од два извора/ two-stream fusion (Yu et al. 2018b)	83.16±0.18	80.22±0.22
вреќа на конволуциски карактеристики/ BoCF (Cheng et al.2017b)	84.32±0.17	82.65±0.31
фино подесување на AlexNet/ fine-tuning AlexNet (Cheng et al. 2017a)	85.16±0.18	81.22±0.19
фино подесување на GoogLeNet/ fine-tuning GoogLeNet (Cheng et al. 2018)	86.02±0.18	82.57±0.12
SAL-TS-Net (Yu et al. 2018a)	87.01±0.19	85.02±0.25
VGG-16+MSCP (Wang et al. 2017)	88.93±0.14	85.33±0.17
фино подесување на VGG-16/ fine-tuning VGG-16 (Cheng et al. 2017a)	90.36±0.18	87.15±0.45
дискриминативни конволуциски мрежи + VGGNet-16/ D-CNN+VGGNet-16 (Cheng et al. 2018)	91.89±0.22	89.22±0.50
триплет мрежи/ triplet networks (Liu et al. 2018b)	92.33±0.20	/
Inception-V3-CapsNet (Zhang W. et al. 2019)	92.60±0.11	89.03±0.21
DenseNet121 со линеарно опаѓање на стапката на учење и линеарна машина со вектори за поддршка (предложениот метод)/ DenseNet121 with linear decay scheduler and linear SVM classifier	93.37±0.12	90.46±0.22
Efficient-B3-aux (Bazi et al.2019)	93.81±0.07	91.08±0.14
„густа“ конволуциска мрежа + 3-D редукација/ Dense-based CNN + 3D pooling (Zhang J. et al. 2019)	94.95±0.04	92.9±0.17



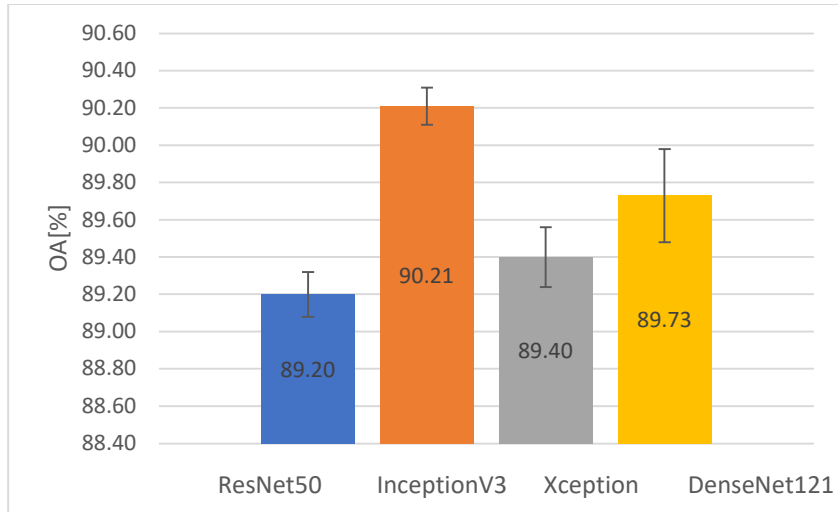
(1)



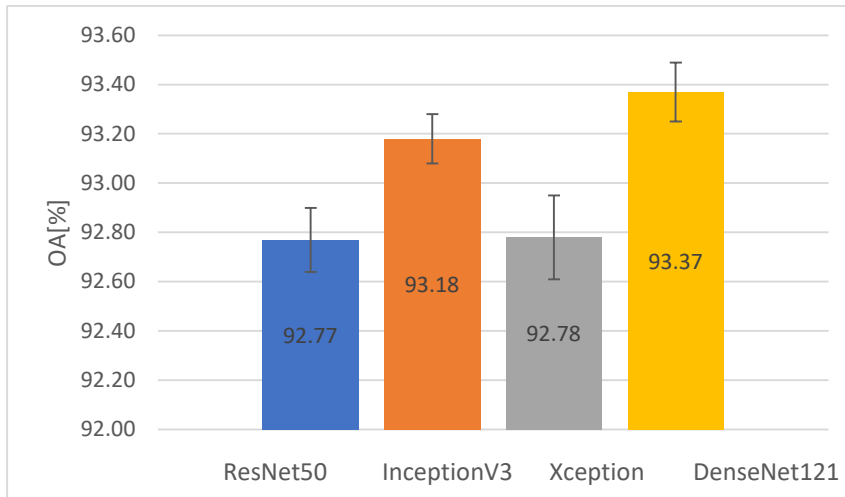
(2)



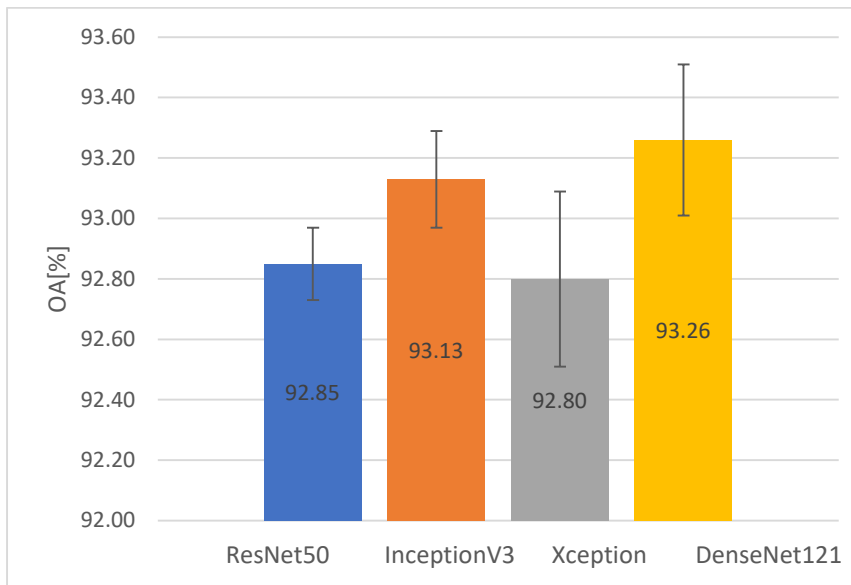
(3)



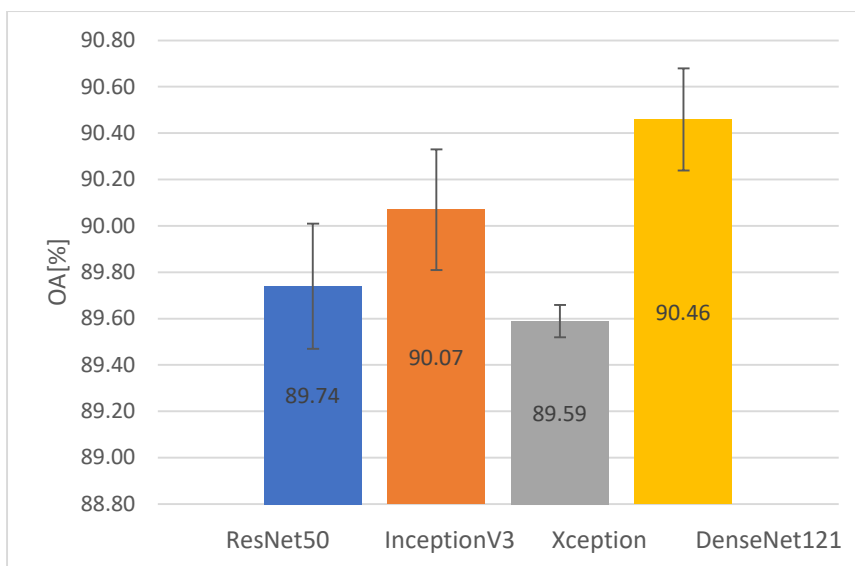
(4)



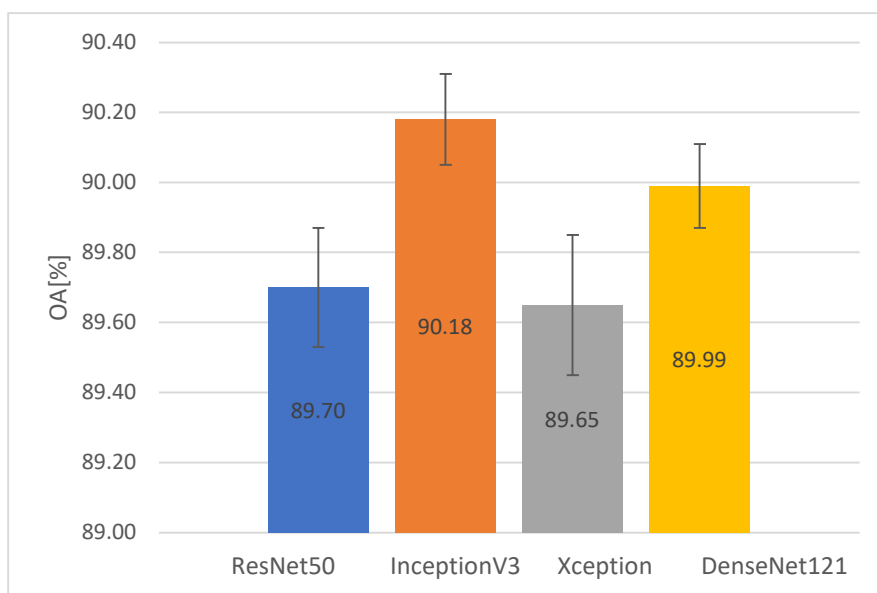
(5)



(6)



(7)



(8)

Слика 5.8 Вкупна точност \pm стандардна девијација (OA \pm STD) на пет повторувања за NWPU-RESISC45 податочното множество со софтмакс класификатор за обем на тренирање, односно тестирање од (1) 20%/80%, линеарно опаѓање на стапката на учење; (2) 20%/80%, циклични стапки на учење; (3) 10%/90%, линеарно опаѓање на стапката на учење, (4) 10%/90%, циклични стапки на учење и со линеарна машина со вектори за поддршка за обем на тренирање, односно тестирање од (5) 20%/80%, линеарно опаѓање на стапката на учење; (6) 20%/80%, циклични стапки на учење; (7) 10%/90%, линеарно опаѓање на стапката на учење, (8) 10%/90%, циклични стапки на учење

Figure 5.8 Overall Accuracy \pm Standard Deviation (OA \pm STD) over five runs for NWPU-RESISC45 data set with softmax classifier under (1) 20%/80% train/test split ratio, linear decay scheduler; (2) 20%/80% train/test split ratio, cyclical learning rates; (3) 10%/90% train/test split ratio, linear decay scheduler, (4) 10%/90% train/test split ratio, cyclical learning rates; and with linear SVM classifier under (5) 20%/80% train/test split ratio, linear decay scheduler; (6) 20%/80% train/test split ratio, cyclical learning rates; (7) 10%/90% train/test split ratio, linear decay scheduler, (8) 10%/90% train/test split ratio, cyclical learning rates

6. ДИСКУСИЈА

Постојат многу моќни техники за класификација на множества на слики снимени од далечина, со кои се постигнува многу голема точност. Целта на овој докторски труд не е да се надмине секоја од нив. Задачата на истражувањата овде е да се придонесе во насоките кои не се истражувани интензивно, како, на пример, одредени типови на претходно тренирани конволуциски мрежи.

6.1 Дискусија за резултатите добиени со методот со екстракција на карактеристики

Од завршените симулации и добиените резултати со методот со екстракција на карактеристики, може да се сумираат следните вредни концепти:

- Наспроти многуте истражувачки студии и теории дека целосно поврзаните нивоа или нивоата за редукција на просек даваат карактеристики кои постигнуваат најголема точност на класификација, ова истражување го сугерира спротивното. Кај класификацијата на податочните множества со слики снимени од далечина преку пренос на учење со екстракција на карактеристики, најголема точност се постигнува со карактеристиките извлечени од средните конволуциски нивоа. Тоа е резултат на фактот дека конволуциските невронски мрежи се оригинално тренирани на сосема различни податочни множества и дека карактеристиките од нивоата за редукција на просек се поврзани со семантичкото значење на сликите. Од друга страна, карактеристиките од средните конволуциски нивоа претставуваат некои основни елементи на сликата како линии, рабови, бои, сенки итн., а се заеднички за различни податочни множества. Посебно внимание треба да се посвети на `mixed_8` конволуциското ниво од `InceptionV3` конволуциската мрежа.

- Класификацијата на воздушни сцени се спроведува со два линеарни класификатори: класификаторот со логистичка регресија и машината со вектори за поддршка даваат слични резултати и на тој начин ги потврдуваат нашите заклучоци. Кога карактеристиките се екстрахираат исклучиво од едно ниво (независно од тоа дали се екстрахираат од нивото за редукција на просек или од некое од конволуциските нивоа), трендовите на зголемување или намалување на точноста на класификација се исти со кој било класификатор. За сите претходно тренирани конволуциски мрежи, освен за `ResNet50`, точноста се

зголемува со придвижување кон средните нивоа. Претставува предизвик да се продолжи понатаму кон пониските ниво и да се спореди точноста на класификација.

- Предложениот метод за класификација на множества од воздушни сцени, базиран на фузија на карактеристики со трансформација со анализа на главните компоненти, води кон точност која е споредлива со современите методи. Меѓутоа, иако постојат методи кои постигнуваат поголема точност на класификација на UC Merced податочното множество, точноста на класификација која се добива на класата „густа станбена област“ (најсложената) е најголема споредено со другите методи презентирани во литературата. Имено, предложениот метод постигнува точност на класификација од 0.88 (88%) за обем на подмножеството за тренирање од 50%, споредено со методот базиран на интеграција на карактеристики со глобален контекст и карактеристики кои се однесуваат на локалните објекти (GCFs +LOFs) со точност на класификација на „густата станбена област“ од 0.74 (74%) и Inception-v3-CapsNet со точност на класификација на „густата станбена област“ од 0.8 (80%). Ова може да најде примена кога станува збор за класификација на податочни множества со големи интер-класни сличности (класи на слики кои лесно може да се измешаат едни со други).

- Предложениот метод со фузија на карактеристики со трансформација со анализа на главните компоненти е конкурентен со други современи методи. Но, треба да се спомене дека тој има подобри перформанси за помал обем на сликите за тренирање. За обем на тренирање од 40% од WHU RS податочното множество, техниката дава точност на класификација од 98.26 ± 0.40 , што ги надминува другите методи во литературата. Ова може да биде корисно во ситуации кога нема можност за долготрајно тренирање на класификаторите, а прифатлива точност на класификација може да се постигне со помал обем на слики за тренирање. Долготрајното тренирање никогаш не е добра идеја, посебно кога има потреба од експерименти кои се повторуваат.

- Во експерименталните поставки се користи и умерено вештачко проширување (data augmentation) на подмножеството за тренирање, што, секако, помага да се постигнат добри резултати на класификација. Вештачкото проширување на подмножеството за тренирање може да биде и поагресивно и да создаде повеќе примероци од оригиналната слика (тука се генерираат пет

примероци од една слика). Тоа се прави со транслации, ротации, промени во размерот, отсекување и хоризонтално и вертикално превртување на сликите од податочното множество. Ова води кон подобрување на точноста на класификација, како резултат на поголемиот обем на податоци за тренирање на линеарните класификатори, што може да биде уште поконкурентно.

- Во извршените експерименти податочното множество се дели на дел за тренирање и дел за тестирање комплетно случајно, со други зборови, не се користи стратификација. Стратифицираната поделба на податоците може да води кон подобри точности на класификација, бидејќи класите на слики кои потешко се разликуваат се еднакво присутни во подмножествата за тренирање и тестирање како и сите други класи. Овој избор е намерен (процесот треба да биде случаен), дури и ако тоа води кон пониска средна вредност на точноста на класификација и поголема стандардна девијација.

Горенаведените согледувања ни даваат вредни информации и насоки за истражување на поконкурентни методи на поголеми и покомплексни податочни множества, за да се обезбеди напредок во класификацијата на воздушни сцени.

Како алтернатива, направен е и обид да се користат карактеристики екстрахирани од модели за длабоко учење и потоа да се изведува традиционална селекција на карактеристики и класификација со други ансамбл-базирани алгоритми (Random Forest, XGBoost, Adaboost, Extremely Randomized Trees). Меѓутоа, податоците кои се експортираат по екстракција на карактеристиките се од огромни размери, па не постои можност за ефективна класификација базирана на рангирање или корелација. Хибридни методи или методи на обвивка (wrapper) се дури и помалку погодни, поради нивната комплексност. Но сепак, сметаме дека ваквата анализа е важна и може да има свој придонес за подобрување на точноста на класификација.

6.2 Дискусија за резултатите добиени со методот со фино подесување

Од комплетираните симулации и експерименталните резултати добиени со методот со фино подесување, може да се донесат неколку заклучоци. Сите презентирани точки, освен последните две, се однесуваат на експериментите со линеарна машина со вектори за поддршка. Последните две точки се однесуваат на случаите кои вклучуваат класификација со машина со вектори за поддршка

со Гаусово јадро. Главните поенти кои произлегуваат од предложениот метод со фино подесување се следните:

- Претходно тренираната InceptionV3 мрежа постигнува најголема точност на класификација кај преносот на учење со фино подесување за AID податочното множество. За NWPU-RESISC45 податочното множество, DenseNet121 постигнува најдобри експериментални резултати, но InceptionV3 е втората најдобра претходно тренирана конволуциска мрежа. Сликите од AID множеството имаат оригинални димензии 600x600, а сликите од NWPU-RESISC45 податочното множество имаат димензии 256x256. Секоја од претходно тренираните конволуциски мрежи бара слики со одредени димензии на својот влез: 299x299 за InceptionV3, 224x224 за DenseNet121. Постигнатите неоптимални резултати можеби зависат од кроењето на сликите од податочното множество на бараните влезни димензии. Земајќи ја предвид постигнатата топ-1 и топ-5 точност на класификација на ImageNet податочното множество, може да се очекува дека Xception би требало да биде конволуциската мрежа со најдобри перформанси, но тоа не е случај. Меѓутоа, InceptionV3 е веднаш зад Xception според постигнатите резултати на ImageNet, па тоа се рефлектира на истражувањата овде.

- Линеарното опаѓање на стапката на учење дава подобра точност на класификација во експерименталните сценарија со обем на тренирање, односно тестирање од 50%/50% за AID податочното множество и класификација со линеарна машина со вектори за поддршка. Цикличните стапки на учење (CLR) се подобри во случаите со 20% обем на сликите за тренирање за AID податочното множество. Двата размера на обемот за тренирање, односно тестирање за NWPU-RESISC45 податочното множество (20%/80% и 10%/90%) даваат мешани резултати за класификација со линеарна машина со вектори за поддршка: половина од нив во корист на линеарното опаѓање на стапката на учење, половина од нив во корист на цикличните стапки на учење. Последните може да претставуваат вистинско решение за експериментални поставки со помал обем на подмножеството за тренирање. Финото подесување на невронските мрежи со циклични стапки на учење резултира во постабилен тренинг и поради тоа помалку склон кон претренирање, споредено со тренирањето со линеарно опаѓање на стапката на учење. Во експериментите овде се користи триаголна шема за цикличните стапки за учење, но опција е и

триаголната 2 шема. Без разлика која шема ќе се имплементира, таа треба да обезбеди вистинска рамнотежа меѓу стабилноста и точноста на тренирањето.

- Во сите симулациони сценарија, имплементирано е измазнување на лабели со фактор=0.1 како метод на регуларизација. Тоа е комбинирано со испуштање (dropout) со фактор=0.5. Регуларизационите техники или комбинација од нив се корисни за борба со претренирањето (overfitting) и за подобрување на способноста на мрежата за генерализација. Целта на предложениот метод е да се зголеми точноста на класификација на множествата на слики снимени од далечина, па од таа причина не се изведувани експериментални сценарија без измазнување на лабели.

- Точноста на класификација која се постигнува со линеарна машина со вектори за поддршка е повисока во повеќето случаи од точноста на класификација добиена со софтмакс нивото. Софтмакс класификаторот функционира подобро за AID податочното множество: тој постигнува подобри експериментални резултати за InceptionV3 и Xception со 50% од податоците наменети за тренирање и за сите конволуциски мрежи освен за ResNet50 со 20% од податоците наменети за тренирање. Линеарната машина со вектори за поддршка е подобра опција за NWPU-RESISC45 податочното множество: таа го надминува софтмакс нивото во сите случаи, освен при фино подесување на InceptionV3 со двата типа стапки на учење и на Xception со линеарно опаѓање на стапката на учење и 10% обем на сликите за тренирање. Може да се заклучи дека екстракцијата на карактеристики од фино подесените конволуциски мрежи резултира во подобра точност на класификација отколку „од-крај-до-крај“ (end-to-end) тренирањето со софтмакс ниво за класификација. Ова е во согласност со презентираниот во табелите 5.13 и 5.18, дека најдобриот метод за класификација на AID и NWPU-RESISC45 податочното множество се базира на екстракција на карактеристики (повеќекратна) и 3D редукација на екстрахираните карактеристики.

- Експерименталното истражување на класификацијата на податочните множества од воздушни сцени со нелинеарна машина со вектори за поддршка, покажува дека тоа е супериорна техника за класификација споредена со софтмакс класификацијата. За AID податочното множество класификацијата со машина со вектори за поддршка со Гаусово јадро ја надминува софтмакс класификацијата во сите случаи, освен за Xception со обем на сликите за

тренирање од 50% и линеарно опаѓање на стапката за учење. Состојбата со NWPU-RESISC45 податочното множество го следи претходниот пример: класификацијата со нелинеарна машина со вектори за поддршка го надминува софтмакс нивото секаде, освен за Xception со обем на сликите за тренирање од 10% и линеарно опаѓање на стапката за учење.

- Споредувајќи го линеарното опаѓање на стапката на учење и цикличните стапки за класификацијата со нелинеарна машина со вектори за поддршка, се доаѓа до слични заклучоци како и кај линеарната машина. Линеарната стапка на учење покажува подобри експериментални резултати со поголеми подмножества за тренирање (50% од AID податочното множество и 20% од NWPU-RESISC45 податочното множество). Цикличните стапки на учење изгледаат попогодни за класификација на множества од воздушни сцени со помал обем на слики за тренирање.

Во анализата на методот со фино подесување за класификација на множества од воздушни сцени повеќе внимание им е посветено на експерименталните постановки за „од-крај-до-крај“ (end-to-end) фино подесување на претходно тренирани конволуциски мрежи, а потоа за екстракција на карактеристики и класификација со линеарна машина со вектори за поддршка. Тоа е разбирливо, бидејќи овие случаи се повторени по пет пати и добиените резултати може да се сметаат за веродостојни. Меѓутоа, и покрај тоа што симулациите кои вклучуваат машина со вектори за поддршка со Гаусово јадро се извршени само по еднаш, воочливи се можностите на ваквата класификација и таа треба понатаму да се експлоатира за развивање на нови супериорни методи за класификација на податочни множества од слики снимени од далечина.

7. ЗАКЛУЧОК

Во овој докторски труд разработени се методи за класификација на слики снимени од далечина со висока резолуција преку пренос на учење (знаење). Првиот предложен метод се базира на спојување на карактеристики од два извора. Во експериментите овде, најпрво претходно тренираните конволуциски мрежи на ImageNet податочното множество се користат за екстракција на карактеристики од оригиналните воздушни сцени од различни нивоа на архитектурата на невронските мрежи. По екстракцијата, карактеристиките извлечени од нивото за редукција на просек и трансформираниите карактеристики од конволуциското ниво со анализа на главните компоненти се спојуваат, за да се формира единствена претстава на репрезентите. На крајот, се користи линеарна машина со вектори за поддршка за класификација на финалното множество на карактеристики. Предложениот метод е тестиран на две податочни множества. Во споредба со други современи методи, оваа архитектура постигнува споредливи резултати. Предложениот метод може да биде релевантен кога е потребно да се тренираат класификатори со мал обем на подмножеството за тренирање. Дополнително, оваа техника може да биде добро решение за класификација на податочни множества со класи на слики со големи интер-класни сличности, како „густа станбена област“ од UC Merced податочното множество. Предложената техника може понатаму да се истражува со екстрахирање на карактеристики од подолните слоеви на претходно тренираните конволуциски мрежи. Како дополнение на ова, стратификацијата може исто така да има влијание на точноста на класификација на истражуваната техника. Така што, за да се зголеми точноста и да се доближи до најдобрите практики при класификација, поделбата на податочното множество на делови за тренирање и тестирање треба да биде стратифицирана. Целото стекнато знаење и искуство би можело да се провери на други податочни множества на слики снимени од далечина, не нужно јавно достапни, по можност со мал обем, бидејќи предложениот метод дава добри резултати со мал дел на слики за тренирање.

Вториот предложен метод за разрешување на проблемот на класификација на множества од воздушни сцени е фино подесување. Кај овој метод најпрво се изведува „од крај до крај“ (end-to-end) тренирање на конволуциските мрежи чии што иницијални тежини не се случајни, туку се

подесени на ImageNet податочното множество. Потоа се изведува екстракција на карактеристики од фино подесените мрежи и истите се класифицираат со машина со вектори за поддршка, со линеарно и со Гаусово јадро. Се покажува дека усвојувањето на линеарно опаѓање на стапката на учење или на циклични стапки на учење, комбинирано со регуларизациони техники како измазнување на лабели, може да доведе до конкурентни резултати во смисла на вкупна точност. Оваа техника е исто така применета на две податочни множества, но овие се од поголем размер. Класификацијата со машина со вектори за поддршка со линеарно јадро дава подобри резултати од софтмакс нивото кога се користи обем за тренирање од 10% и 20% од NWPU-RESISC45 податочното множество, споредено со резултатите при обем на тренирање од 50% и 20% од AID податочното множество. Ваквиот исход е очекуван, бидејќи познато е дека машината со вектори за поддршка е поробусна во случаи на помало подмножество за тренирање (Shao et al. 2012). Дискусијата на методот со фино подесување ни дава вредни информации при истражување на поконкурентни методи со цел да се обезбеди напредок во класификацијата на податочни множества од воздушни сцени. Можни насоки на понатамошно проучување се: да се испитува методот на фино подесување со различни типови на претходно тренирани конволуциски невронски мрежи со различни видови на архитектури, да се вклучи пронаоѓач на стапката на учење (Smith 2017) со цел да се одредат оптималните граници за цикличните стапки на учење или иницијалната стапка на учење кај линеарното опаѓање на истата, да се направи обид за подобрување на резултатите со фино подесување на само некои слоеви на претходно тренираните конволуциски мрежи, наспроти одмрзнувањето на целата мрежна архитектура за тренирање и да се воведат стратифицирана поделба на подмножествата за тренирање и тестирање. Предложената техника за фино подесување, заедно со насоките за нејзино унапредување, би можела да се примени при класификација на множества од воздушни сцени од поголеми размери.

8.КОРИСТЕНА ЛИТЕРАТУРА (REFERENCES)

1. Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., & Devin M. (2016). Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv:1603.04467
2. Anwer R.M., Khan F.S., vandeWeijer J., Monlinier M., & Laaksonen J. (2017) Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. arXiv:1706.01171
3. Bao W., Yue J., & Rao Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE, 12, e0180944
4. Bazi Y., Rahhal M.M.A., Alhichri H., & Alajlan N. (2019). Simple Yet Effective Fine-Tuning of Deep CNNs Using an Auxiliary Classification Loss for Remote Sensing Scene Classification. Remote Sens., 11, 2908.
5. Bengio Y. (2012). Practical recommendations for gradient-based training of deep architectures. In: Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_26
6. Bergstra J., & Bengio Y. (2012). Random search for hyper-parameter optimization. The Journal of Machine Learning Research, 13(10):281–305
7. Bian X., Chen C., Tian L., Du Q. (2017). Fusing local and global features for high-resolution scene classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., 10, 2889–2901
8. Blei M.D., Ng, A.Y., & Jordan M.I. (2003). Latent Dirichlet allocation. J. Mach. Learn. Res., 3, 993–1022
9. Bosch A., Zisserman A., & Muñoz X. (2006). Scene classification via pLSA. In Proceedings of the European Conference on Computer Vision, Graz, pp. 517–530
10. Castelluccio M., Poggi G., Sansone C. & Verdoliva L. (2015). Land use classification in remote sensing images by convolutional neural networks, arXiv:1508.00092
11. Chaib S., Liu H., Gu Y., & Yao H. (2017). Deep feature fusion for VHR remote sensing scene classification. IEEE Trans. Geosci. Remote Sens., 55, 4775–4784
12. Chatfield K., Simonyan K., Vedaldi A., & Zisserman A. (2014) Return Devil Details: Delving Deep Convolutional Nets., arXiv:1405.3531

13. Chen L., Yang W., Xu K., & Xu T. (2011). Evaluation of local features for scene classification using VHR satellite images. In Proceedings of the Joint Urban Remote Sensing Event (JURSE), Munich, Germany, pp. 385–388.
14. Chen C., Zhang B., Su H., Li W., & Wang L. (2015). Land-use scene classification using multi-scale completed local binary patterns. *Signal Image and Video Process.*, 10, 745–752
15. Chen S., & Tian Y. (2015). Pyramid of spatial relations for scene-level land use classification. *IEEE Trans. Geosci. Remote Sens.*, 53, 1947–1957
16. Chen Y., Xu W., Zuo, J., & Yang K. (2018). The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier. *Clust. Comput.* 2018, 7665–7675
17. Chen J., Wang C., Ma Z., Chen J., He D., & Ackland S. (2018). Remote sensing scene classification based on convolutional neural networks pre-trained using attention-guided sparse filters. *Remote Sens.*, 10, 290
18. Cheng G., Han J., & Lu X. (2017). Remote Sensing Image Classification: Benchmark and State of the Art. *Proc. IEEE*, 105, 1865–1883
19. Cheng G., Li Z., Yao X., Li K., & Wei Z. (2017). Remote Sensing Image Scene Classification Using Bag of Convolutional Features. *IEEE Geosci. Remote Sens. Lett.*, 14
20. Cheng G., Yang C., Yao X., Guo L., & Han J. (2018) When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.*, 56, 2811–2821
21. Cheng G., Xie X., Han J., Guo L., & Xia G. (2020) Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *arXiv:2005.01094*
22. Chollet F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions., *arXiv:1610.02357v3*
23. Corizzo R., Ceci M., & Japkowicz N. (2019). Anomaly Detection and Repair for Accurate Predictions in Geo-distributed Big Data. *Big Data Res.*, 16, 18–35
24. Corizzo R., Ceci M., Zdravevski E., & Japkowicz, N. (2020). Scalable auto-encoders for gravitational waves detection from time series data. *Expert Syst. Appl.*, 151, 113378
25. Dauphin Y.N., de Vries, H., Chung J., & Bengio Y. (2015). Rmsprop and equilibrated adaptive learning rates for non-convex optimization. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada

26. Deng J., Dong W., Socher R., Li L.J., Li K., & Fei-Fei L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA
27. dos Santos J.A., Penatti, O.A.B., & da Silva Torres R. (2010). Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In Proceedings of the VISAPP (2), Angers, France, pp. 203–208.
28. Dozat T. (2015). Incorporating Nesterov Momentum into Adam. http://cs229.stanford.edu/proj2015/054_report.pdf
29. Duan L., Xu D., & Tsang I.W. (2012) "Learning with Augmented Features for Heterogeneous Domain Adaptation". IEEE Trans Pattern Anal Mach Intell, 36(6):1134–48.
30. Duchi J., Hazan E., & Singer Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In: J. Mach. Learn. Res. 12, pages 2121– 2159. ISSN: 1532-4435
31. Gan J., Li Q., Zhang Z., & Wang J. (2016). Two-level feature representation for aerial scene classification. IEEE Geosci. Remote Sens. Lett., 13, 1626–1630
32. Girshick R., Donahue J., Darrell T., & Malik J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Computer Vision and Pattern Recognition, IEEE, pp. 580–587
33. Gong X., Xie Z., Liu Y., Shi X., Zheng Z. (2018). Deep salient feature-based anti-noise transfer network for scene classification of remote sensing imagery. Remote Sens., 10, 410
34. Goodfellow I., Bengio Y., & Courville A. (2016). Deep Learning. <http://www.deeplearningbook.org>. MIT Press
35. Guo C., Pleiss G., Sun Y., & Weinberger K.Q. (2017). On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, NSW, Australia, pp. 1321–1330
36. Haralick R.M., Shanmugam K., & Dinstein I.H. (1973). Textural features for image classification. IEEE Trans. Syst. Man Cybern. 610–621, doi:10.1109/TSMC.1973.4309314
37. Harel M., & Mannor S. (2011). Learning from Multiple Outlooks. Proceedings of the 28th international conference on machine learning, p. 401–8
38. He K., Zhang X., Ren S., & Sun J. (2015). Deep Residual Learning for Image Recognition., arXiv:1512.03385v1

39. He N., Fang L., Li S., Plaza A., & Plaza J. (2018). Remote Sensing Scene Classification Using Multilayer Stacked Covariance Pooling. *IEEE Trans. Geosci. Remote Sens.*, 56, 6899–6910
40. Hebb D.O. (1946). *The organization of behaviour; a neuropsychological theory*. NY: Wiley
41. Hinton G., Srivastava N., Krizhevsky A., Sutskever I., & Salakhutdinov R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs.NE]*
42. Hinton G. (2020) *Neural Networks for Machine Learning*. Преземено во октомври 2020. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
43. Hu F., Xia G.-S., Hu J., & Zhang L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.*, 7, 14680–14707
44. Huang F., & Yan L. (2015). Hull vector-based incremental learning of hyperspectral remote sensing images. *J. Appl. Remote Sens.*, 9, 096022.
45. Huang L., Chen C., Li W., & Du Q. (2016). Remote sensing image scene classification using multi-scale completed local binary patterns and fisher vectors. *Remote Sens.*, 8, 483
46. Huang G, Liu Z., van der Maaten L., & Weinberger K.Q. (2018). *Densely Connected Convolutional Networks.*, *arXiv:1608.06993v5*
47. Ioffe S., & Szegedy C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *CoRR abs/1502.03167*
48. Jarrett K., Kavukcuoglu K., Ranzato M., & LeCun Y. (2009). What is the best multi-stage architecture for object recognition? In *Proceeding of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan*, pp. 2146–2153
49. Jégou H., Perronnin F., Douze M., Sanchez J., Perez P., & Schmid C. (2012). Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34, 1704–1716
50. Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., & Darrell T. (2014). *Caffe: Convolutional architecture for fast feature embedding*. In *Proceedings of the ACM International Conference on Multimedia, Orlando, FL, USA*, pp. 675–678

51. Khelifi L., & Mignotte M., (2020). Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis. *IEEE Access* 8, 126385–126400
52. Kingma D.P., & Ba J. (2014). Adam: A Method for Stochastic Optimization. In: *CoRR* abs/1412.6980
53. Krizhevsky A., Sutskever I., & Hinton G.E. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, pp. 1106–1114
54. Kusumaningrum R., Wei H., Manurung R., & Murni A. (2014). Integrated visual vocabulary in latent Dirichlet allocation–based scene classification for IKONOS image. *J. Appl. Remote Sens.*, 8, 083690–083690
55. Lameski P., Zdravevski E., Mingov R., & Kulakov A. (2015). SVM Parameter Tuning with Grid Search and Its Impact on Reduction of Model Over-fitting. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing; Lecture Notes in Computer Science*; Yao Y., Hu Q., Yu H., & Grzymala-Busse J., Eds; Springer: Cham, Germany, Volume 943
56. Larochelle H., Bengio Y., Louradour J., & Lamblin P. (2009). Exploring strategies for training deep neural networks. *J. Mach. Learn. Res.*, 1–40
57. Lazebnik S., Schmid C., & Ponce J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Proc. IEEE Conference on Computer Vision and Pattern Recognition*, New York, NY, pp. 2169–2178
58. LeCun Y., Bottou L., Orr G., & Müller K.R. (1998). Efficient BackProp. *Neural networks: Tricks of the trade*, 9-48
59. Lecun Y., Bottou L., Bengio Y., & Haffner P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Volume: 86, Issue: 11, Nov. 1998), page(s): 2278 – 2324
60. Liang Y., Monteiro S.T., Saber E.S. (2016). Transfer Learning for High-Resolution Aerial Image Classification. In *Proceedings of the IEEE Applied Imagery Pattern Recognition (AIPR) Workshop*, Washington, DC, USA
61. Li R., Wang S. (2018). Adaptive Graph Convolutional Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA

62. Liu Y., Zhong Y., Fei F., Zhu Q., & Qin Q. (2018). Scene Classification Based on a Deep Random-Scale Stretched Convolutional Neural Network. *Remote Sens.*, 10, 444
63. Liu Y., & Huang C. (2018). Scene classification via triplet networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 509
64. Lowe D.G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60, 91–110
65. Lu Y., Luo L., Huang D., Wang, Y., & Chen, L. (2020). Knowledge Transfer in Vision Recognition. *ACM Comput. Surv.*, 53, 1–35
66. Luo B., Aujol J.-F., Gousseau Y., & Ladjal S. (2008). Indexing of satellite images with different resolutions by wavelet features. *IEEE Trans. Image Process.*, 17, 1465–1472
67. Luo B., Aujol J.-F., & Gousseau Y. (2009). Local scale measure from the topographic map and application to remote sensing images. *Multiscale Modeling Simul.* 8, 1–29
68. Luo B., Jiang S., & Zhang L. (2013). Indexing of remote sensing images with different resolutions by multiple features. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 6, 1899–1912
69. Luus F.P.S., Salmon B.P., VanDenBergh F, & Maharaj, B.T.J. (2015). Multi-view deep learning for land-use classification. *IEEE Geosci. Remote Sens. Lett.*, 12, 2448–2452
70. Mahdianpari M., Salehi B., Rezaee M., Mohammadimanesh F., & Zhang Y. (2018). Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery. *Remote Sens.*, 10, 1119
71. Manjunath B.S., & Ma W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.* 1996, 18, 837–842.
72. McCulloch W.S., & Pitts W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115
73. Mignone P., Pio G., D'Elia D., Ceci M. 2019. Exploiting transfer learning for the reconstruction of the human gene regulatory network. *Bioinformatics*, 36, 1553–1561
74. Mishkin D. (2020). CaffeNet-Benchmark – Batch Norm. Преземено во октомври 2020. <https://github.com/duchaaiki/caffenet-benchmark/blob/master/batchnorm.md>
75. Müller R., Kornblith S., & Hinton G.E. (2019). When does label smoothing help? In *Proceedings of the Advances in Neural Information Processing Systems 32: Annual*

Conference on Neural Information Processing Systems, NeurIPS 2019, Vancouver, BC, Canada, pp. 4696–4705

76. Nam J., & Kim S. (2015). Heterogeneous Defect Prediction. Proceedings of the 2015 10th joint meeting on foundations of software engineering, p. 508–19

77. Negrel R., Picard D., & Gosselin P.-H. (2014). Evaluation of second-order visual features for land-use classification. In Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI), Klagenfurt, Austria, pp. 1–5

78. Nesterov Y. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In: Soviet Mathematics Doklady, Volume 27. 2. 1983, pages 372–376

79. Nielsen M. A. (2015). Neural Networks and Deep learning. Determination Press

80. Nogueira K., Penatti O.A.B., & dos Santos J.A., (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognit., 61, 539–556

81. Ojala T., Pietikainen M., & Maenpaa T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell., 24, 971–987

82. Oliva A., & Torralba A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vis., 42, 145–175

83. Pan S.J., & Yang Q. (2010). A Survey on Transfer Learning. IEEE Trans Knowl Data Eng, 22(10):1345–59

84. Penatti O.A.B., Nogueira K., & DosSantos J.A. (2015). Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'15), 44–51, IEEE, Boston, MA, USA

85. Pereyra G., Tucker G., Chorowski J., Kaiser L., & Hinton G.E. (2017). Regularizing Neural Networks by Penalizing Confident Output Distributions. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France

86. Perronnin F., Sanchez J., & Mensink T. (2010). Improving the fisher kernel for large-scale image classification. In Proceedings of the European Conference on Computer Vision, Heraklion, Greece, pp. 143–156

87. Pio G., Ceci M., Prisciandaro F., & Malerba D. (2020). Exploiting causality in gene network reconstruction based on graph embedding. Mach. Learn., 109, 1231–1279

88. Qayyum A., Malik, A.S., Saad, N.M., Iqbal M., Abdullah M.F., & Rasheed W. (2017). Scene classification for aerial images based on CNN using sparse coding technique. *Int. J. Remote Sens.*, 38, 2662–2685
89. Qi K., Guan Q., Yang C., Peng F., Shen S., & Wu H. (2018). Concentric Circle Pooling in Deep Convolutional Networks for Remote Sensing Scene Classification. *Remote Sens.*, 10, 934
90. Qian N. (1999). On the Momentum Term in Gradient Descent Learning Algorithms. In: *Neural Netw.* pages 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6
91. Razavian A.S., Azizpour H., Sullivan J., & Carlsson S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, Columbus, OH, USA, p. 512519
92. Reddit community contributors. (2020). Batch Normalization before or after ReLU? Преземено во октомври 2020. https://www.reddit.com/r/MachineLearning/comments/67gonq/d_batch_normalization_before_or_after_relu/
93. Risojevic V., & Babic, Z. (2011). Aerial image classification using structural texture similarity. In *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Bilbao, Spain, pp. 190–195.
94. Risojevic V., & Babic Z. (2012). Orientation difference descriptor for aerial image classification. In *Proceedings of the International Conference on Systems, Signals and Image Processing (IWSSIP)*, Vienna, Austria, pp. 150–153.
95. Risojevic V., & Babic Z. (2013). Fusion of global and local descriptors for remote sensing image classification. *IEEE Geosci. Remote Sens. Lett.*, 10, 836–840
96. Rosenblatt F. (1957). *The perceptron: A Perceiving and Recognizing Automation*. Report 85-60-1, Cornell Aeronautical Laboratory, Buffalo, New York
97. Rosenblatt F. (1961) *Principles of neurodynamics: Perceptrons and the theory of brain mechanism*. Spartan Books. Washington, DC
98. Ruder S. (2016). An overview of gradient descent optimization algorithms. In: *CoRR abs/1609.04747*
99. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A., & Fei-Fei L. (2015). ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.*, 115, 211–252

100. Scheirer W.J., Kumar N., Belhumeur P.N., & Boult T.E. (2012). Multi-attribute spaces: Calibration for attribute fusion and similarity search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 2933–2940
101. Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., & LeCun Y. (2013). Overfeat: Integrated recognition, localization, and detection using convolutional networks. arXiv:1312.6229
102. Shao Y., & Lunetta R.S. (2012). Comparison of support vector machine, neural network, and CART algorithms for the land-cover classification using limited training data points. *ISPRS Journal of Photogrammetry and Remote Sensing* 70, 78 – 87
103. Sheng G., Yang W., Xu T., & Sun H. (2012). High-resolution satellite scene classification using a sparse coding-based multiple feature combination. *Int. J. Remote Sens.*, 33, 2395–2412
104. Shorten C., & Khoshgoftaar T.M. (2019). A survey on Image Data Augmentation for Deep Learning. *J Big Data* 578, 6, 60
105. Simard P.Y., Steinkraus D., & Platt J.C. (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition
106. Sivic J., & Zisserman A. (2003). Video google: A text retrieval approach to object matching in videos. In Proceedings of the IEEE International Conference on Computer Vision, Nice, France, pp. 1470–1477
107. Smith L.N. (2017). Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, IEEE: Piscataway, NJ, USA, pp. 464–472
108. Snoek J., Larochelle H., & Adams R. (2012). Practical Bayesian optimization of machine learning algorithms. Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2: 2951–2959
109. Stricker M.A., & Orengo M. (1995). Similarity of color images. In IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology. International Society for Optics and Photonics: Bellingham, WA, USA, pp. 381–392
110. Sun H., Li S., Zheng X., & Lu X. (2019). Remote Sensing Scene Classification by Gated Bidirectional Network, *IEEE Trans. Geosci*
111. Sutskever I. (2013). Training Recurrent Neural Networks". AAINS22066. PhD thesis. Toronto, Ont., Canada, Canada, ISBN: 978-0-499-22066-0

112. Swain M.J., & Ballard D.H. (1991). Color indexing. *Int. J. Comput. Vis.*7, 11–32
113. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., & Rabinovich A. (2014). Going deeper with convolutions. *arXiv:1409.4842*
114. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., & Wojna Z. (2015). Rethinking the Inception Architecture for Computer Vision., *arXiv:1512.0567v3*
115. Wang C., & Mahadevan S. (2011). Heterogeneous Domain Adaptation Using Manifold Alignment. *Proceedings of the 22nd international joint conference on artificial intelligence*, vol.2, p. 541–46.
116. Wang G., Fan B., Xiang S., & Pan C. (2017). Aggregating Rich Hierarchical Features for Scene Classification in Remote Sensing Imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 10, 4104–4115
117. Wang Q., Liu S., Chanussot J., & Li X. (2018). Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 99, 1155–1167
118. Xia G.S., Yang W., Delon J., Gousseau Y., Sun H., & Mante H. (2010). Structural high-resolution satellite image indexing. In *Proceedings of the ISPRS TC VII Symposium-100 Years ISPRS 38*, Vienna, Austria, pp. 298–303
119. Xia G.S., Hu J., Hu F., & Shi B. (2017). AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.*, 55, 3965–3981
120. Xie M., Jean N., Burke M., Lobell D., & Ermon S. (2015). Transfer learning from deep features for remote sensing and poverty mapping, *arXiv:1510.00098*
121. Yang Y., & Newsam S. (2008). Comparing sift descriptors and Gabor texture features for classification of remote sensed imagery. In *Proceedings of the IEEE International Conference on Image Processing*, San Diego, CA, USA, pp. 1852–1855
122. Yang Y., & Newsam S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA, USA, pp. 270–279.
123. Yang Y., & Newsam S. (2011). Spatial pyramid co-occurrence for image classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, pp. 1465–1472

124. Yang W., Yin X., & Xia G.S. (2015). Learning high-level features for satellite image classification with limited labeled samples. *IEEE Trans. Geosci. Remote Sens.*, 53, 4472–4482.
125. Yosinski J., Clune J., Bengio Y., & Lipson H. (2014). How transferable are features in deep neural networks? In *Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, Montreal, QC, Canada, pp. 3320–3328
126. Yu Y.L., & Liu F.X. (2018). Dense connectivity based two-stream deep feature fusion framework for aerial scene classification. *Remote Sens.*, 10, 1158
127. Yu Y., & Liu F. (2018). A two-stream deep fusion framework for high-resolution aerial scene classification. *Comput. Intell. Neurosci.*, 13
128. Yu Y., & Liu F. (2018). Aerial Scene Classification via Multilevel Fusion Based on Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens.*, 15
129. Yue J., Zhao W., Mao S., & Liu H. (2015). Spectral–spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.*, 6, 468–477
130. Zeiler M.D. (2012) ADADELTA: An Adaptive Learning Rate Method”. In: CoRR abs/1212.5701
131. Zhang J., Lu C., Li X., Kim H.J., & Wang J. (2019). A full convolutional network based on DenseNet for remote sensing scene classification. *Mathematical Bioscience and Engineering*
132. Zhang W., Tang P., & Zhao L. (2019). Remote Sensing Image Scene Classification Using CNN-CapsNet. *Remote Sens.*, 11, 494
133. Zeng D., Chen S., Chen B., & Li S. (2018). Improving remote sensing scene classification by integrating global-context and local-object features. *Remote Sens.*, 10, 734
134. Zhong Y., Cui M., Zhu Q., & Zhang L. (2015). Scene classification based on multi-feature probabilistic latent semantic analysis for high spatial resolution remote sensing images. *J. Appl. Remote Sens.*, 9, 095064
135. Zhou J.T., Pan S., Tsang I.W., & Yan Y. (2014). Hybrid Heterogeneous Transfer Learning Through Deep Learning. *Proceedings of the national conference on artificial intelligence*, vol. 3, p. 2213–20

136. Zhu Y., Chen Y., Lu Z., Pan S., Xue G., Yu Y., & Yang Q. (2011). Heterogeneous Transfer Learning for Image Classification. Proceedings of the national conference on artificial intelligence, vol. 2, p. 1304–9
137. Zhu Q., Zhong Y., Zhao B., Xia G., & Zhang L. (2016). Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. IEEE Geosci. Remote Sens. Lett., 13, 747–751

9. ДОДАТОК - користени кратенки

Remote Sensing	RS
Convolutional Neural Network	CNN
Support Vector Machine	SVM
Radial Basis Function	RBF
Threshold Logic Unit	TLU
Stochastic Gradient Descent	SGD
Adaptive Moment Estimation	Adam
Rectified Linear Unit	ReLU
Exponential Linear Unit	ELU
Batch Normalization	BN
Remote Sensing	RS
Scale Invariant Feature Transform	SIFT
Grey Level Co-Occurrence Matrix	GLCM
Enhanced Gabor Texture Descriptor	EGTD
Bag of Visual Words	BoVW
Local Binary Pattern	LBP
Principal Component Analysis	PCA
Spatial Pyramid Co-occurrence Kernel	SPCK
Spatial Pyramid Match kernel	SPM
Spatial Co-occurrence Kernel	SCK
Latent Dirichlet Allocation	LDA
Edge Orientation Histogram	EOH
Improved Fisher Kernel	IFK
Red Green Blue	RGB

Recurrent Neural Network	RNN
Logistic Regression Classifier	LRC
Radial Basis Function	RBF
Dense Convolutional Network	DenseNet
Maximum Likelihood Estimation	MLE
Cyclical Learning Rates	CLR
Overall Accuracy	OA
Global Context Features	GCF
Local Object Features	LOF
Aerial Image Dataset	AID
Northwestern Polytechnical University – Remote Sensing Image Scene Classification	NWPU-RESISC45
Modified National Institute of Standards And Technology	MNIST
Extreme Value Theory	EVT
Vectors of Locally Aggregated Tensors	VLAT
Vector of Locally Aggregated Descriptors	VLAD
ImageNet Large Scale Visual Recognition Challenge	ILSVRC
Discriminant Correlation Analysis	DCA