



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Никола Савић

**Примена конволуционих неуронских
мрежа у класификацији листова биљака
према здравственом стању**

ДИПЛОМСКИ РАД
- Основне академске студије -

Нови Сад, 2022.



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
21000 НОВИ САД, Трг Доситеја Обрадовића 6

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА


Редни број, РБР:	
Идентификациони број, ИБР:	
Тип документације, ТД:	монографска публикација
Тип записа, ТЗ:	текстуални штампани документ
Врста рада, ВР:	дипломски (Bachelor) рад
Аутор, АУ:	Никола Савић
Ментор, МН:	Владимир Иванчевић
Наслов рада, НР:	Примена конволуционих неуронских мрежа у класификацији листова биљака према здравственом стању
Језик публикације, ЈП:	српски
Језик извода, ЈИ:	српски
Земља публиковања, ЗП:	Србија
Уже географско подручје, УГП:	Војводина
Година, ГО:	2022
Издавач, ИЗ:	ауторски репринт
Место и адреса, МА:	Нови Сад, Факултет техничких наука, Трг Доситеја Обрадовића 6
Физички опис рада, ФО: (поглавља/страна/ цитата/табела/слика/графика/прилога)	9/44/36/7/9/0/0
Научна област, НО:	Интердисциплинарно (Електротехничко и рачунарско инжењерство; Индустриско инжењерство и инжењерски менаџмент)
Научна дисциплина, НД:	Примењене рачунарске науке и информатика
Предметна одредница/Кључне речи, ПО:	машинско учење, конволуционе неуронске мреже, класификација слика
УДК	
Чува се, ЧУ:	Библиотека Факултета техничких наука, Трг Доситеја Обрадовића 6, Нови Сад
Важна напомена, ВН:	
Извод, ИЗ:	У овом раду представљен је комплетан циклус класификације слика конволуционим неуронским мрежама. Дат је теоријски опис коришћених поступака и метода, описани су коришћени скупови података, приказан је процес предобrade података и обуке модела класификације, и дискутовано је о добијеним резултатима модела. На крају, дат је закључак о приложеним резултатима и дискутовано је о могућем будућем раду на коришћеним скуповима података.
Датум прихватања теме, ДП:	
Датум одбране, ДО:	
Чланови комисије, КО:	Председник: др Дину Драган, ванредни професор
	Члан: др Милан Челиковић, доцент
	Члан, ментор: др Владимир Иванчевић, ванредни професор
	Потпис ментора

Образац **Q2.HA.04-05** - Издање 1



KEY WORDS DOCUMENTATION

Accession number, ANO :			
Identification number, INO :			
Document type, DT :	monographic publication		
Type of record, TR :	text printed document		
Contents code, CC :	Bachelor thesis		
Author, AU :	Nikola Savić		
Mentor, MN :	Vladimir Ivančević		
Title, TI :	Application of convolutional neural networks in classification of plant leaves by health status		
Language of text, LT :	Serbian		
Language of abstract, LA :	Serbian		
Country of publication, CP :	Serbia		
Locality of publication, LP :	Vojvodina		
Publication year, PY :	2022		
Publisher, PB :	author's reprint		
Publication place, PP :	Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6		
Physical description, PD : (chapters/pages/ref./tables/pictures/graphs/appendixes)	9/44/36/7/9/0/0		
Scientific field, SF :	Interdisciplinary (Electrical and Computer Engineering; Industrial Engineering and Engineering Management)		
Scientific discipline, SD :	Applied Computer Science and Informatics		
Subject/Key words, S/KW :	machine learning, convolutional neural networks, image classification		
UC			
Holding data, HD :	Library of the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad		
Note, N :			
Abstract, AB :	This thesis presents a complete cycle of image classification by convolutional neural networks. A theoretical description of the used procedures and methods is given, the used data sets are described, the process of data preprocessing and training of the classification model is presented, and the obtained model results are discussed. Finally, a summary of the attached results is given and possible future work on the used data sets is discussed.		
Accepted by the Scientific Board on, ASB :			
Defended on, DE :			
Defended Board, DB :	President:	Dinu Dragan, Ph.D., associate professor	
	Member:	Milan Čeliković, Ph.D., assistant	Menthor's sign
	Member, Mentor:	Vladimir Ivančević, Ph.D., associate professor	

	УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА 21000 НОВИ САД, Трг Доситеја Обрадовића 6	Датум:
	ЗАДАТАК ЗА ИЗРАДУ ДИПЛОМСКОГ (BACHELOR) РАДА	Лист/Листова:
		1/1

(Податке уноси предметни наставник - ментор)

Врста студија:	<input checked="" type="checkbox"/> Основне академске студије
Студијски програм:	Информациони инжењеринг
Руководилац студијског програма:	др Славица Кордић, ванредни професор

Студент:	Никола Савић	Број индекса:	IN 35/2018
Област:	Интердисциплинарно (Електротехничко и рачунарско инжењерство; Индустрijско инжењерство и инжењерски менаџмент)		
Ментор:	др Владимир Иванчевић, ванредни професор		

НА ОСНОВУ ПОДНЕТЕ ПРИЈАВЕ, ПРИЛОЖЕНЕ ДОКУМЕНТАЦИЈЕ И ОДРЕДБИ СТАТУТА ФАКУЛТЕТА ИЗДАЈЕ СЕ ЗАДАТАК ЗА ДИПЛОМСКИ (Bachelor) РАД, СА СЛЕДЕЋИМ ЕЛЕМЕНТИМА:

- проблем – тема рада;
- начин решавања проблема и начин практичне провере резултата рада, ако је таква провера неопходна;
- литература

НАСЛОВ ДИПЛОМСКОГ (BACHELOR) РАДА:

ПРИМЕНА КОНВОЛУЦИОНИХ НЕУРОНСКИХ МРЕЖА У КЛАСИФИКАЦИЈИ ЛИСТОВА БИЉАКА ПРЕМА ЗДРАВСТВЕНОМ СТАЊУ

ТЕКСТ ЗАДАТКА:

<p>Над одабраним подацима обучити моделе машинског учења у облику конволуционих неуронских мрежа који би могли подржати поступак класификације листова биљака према здравственом стању. При томе, испитати могућности више различитих архитектура конволуционих неуронских мрежа. Спровести детаљнију анализу успешности за одабране моделе машинског учења. Коришћене податке по потреби додатно припремити пре спровођења обучавања. Приликом примене софтверских технологија примарно се ослањати на одговарајућу званичну документацију.</p>
--

Руководилац студијског програма:	Ментор рада:

Примерак за: <input type="checkbox"/> - Студента; <input type="checkbox"/> - Ментора
--

Образац Q2.НА.15-04 - Издање 2

Садржај

1. УВОД	7
2. КОНВОЛУЦИОНЕ НЕУРОНСКЕ МРЕЖЕ	8
2.1 ВЕШТАЧКИ НЕУРОН	8
2.2 АКТИВАЦИОНЕ ФУНКЦИЈЕ	8
2.3 СТРУКТУРА НЕУРОНСКИХ МРЕЖА	9
2.4 ФУНКЦИЈА ГУБИТКА	10
2.5 ОБУЧАВАЊЕ КРОЗ ЕПОХЕ	10
2.6 ПРЕКОМЕРНО ПРИЛАГОЂАВАЊЕ	11
3. КЛАСИФИКАЦИЈА СЛИКА	12
3.1 КОРИШЋЕНЕ ТЕХНОЛОГИЈЕ	12
3.2 ВАЛИДАЦИЈА	13
3.3 МЕРЕ УСПЕШНОСТИ КЛАСИФИКАТОРА	13
3.4 АРХИТЕКТУРЕ КОНВОЛУЦИОНЕ НЕУРОНСКЕ МРЕЖЕ	15
3.4.1 АРХИТЕКТУРА <i>LENET5</i>	15
3.4.2 АРХИТЕКТУРА <i>ALEXNET</i>	17
3.4.3 АРХИТЕКТУРА <i>VGG16</i>	18
3.4.4 АРХИТЕКТУРА <i>DENSENET</i>	19
4. СКУПОВИ ПОДАТАКА	20
4.1 СКУП ПОДАТАКА <i>IBEAN</i>	20
4.2 СКУП ПОДАТАКА <i>PLANTAЕК</i>	21
4.3 ПРЕДОБРАДА ПОДАТАКА	22
4.4 УЧИТАВАЊЕ ПОДАТАКА	23
5. ОБУКА МОДЕЛА	24
5.1 МОДЕЛ СА АРХИТЕКТУРОМ <i>LENET5</i>	25
5.1.1 ОБУКА ЗА СКУП ПОДАТАКА <i>IBEAN</i>	26
5.1.2 ОБУКА ЗА СКУП ПОДАТАКА <i>PLANTAЕК</i>	26
5.2 МОДЕЛ СА АРХИТЕКТУРОМ <i>ALEXNET</i>	27
5.2.1 ОБУКА ЗА СКУП ПОДАТАКА <i>IBEAN</i>	28
5.2.2 ОБУКА ЗА СКУП ПОДАТАКА <i>PLANTAЕК</i>	28
5.3 МОДЕЛ СА АРХИТЕКТУРОМ <i>VGG16</i>	29
5.3.1 ОБУКА ЗА СКУП ПОДАТАКА <i>IBEAN</i>	30
5.3.2 ОБУКА ЗА СКУП ПОДАТАКА <i>PLANTAЕК</i>	31
5.4 МОДЕЛ СА АРХИТЕКТУРОМ <i>DENSENET</i>	31
5.4.1 ОБУКА ЗА СКУП ПОДАТАКА <i>IBEAN</i>	32
5.4.2 ОБУКА ЗА СКУП ПОДАТАКА <i>PLANTAЕК</i>	32
6. РЕЗУЛТАТИ И ДИСКУСИЈА	33
6.1 ПЕРФОРМАНСЕ МОДЕЛА	33
6.1.1 ПЕРФОРМАНСЕ НА СКУПУ ПОДАТАКА <i>IBEAN</i>	33
6.1.2 ПЕРФОРМАНСЕ НА СКУПУ ПОДАТАКА <i>PLANTAЕК</i>	34
6.2 ВРЕМЕНА ПОТРЕБНА ЗА ОБУКУ МОДЕЛА	35

6.3	ОДАБИР НАЈБОЉЕГ МОДЕЛА.....	35
7.	ЗАКЉУЧАК	39
8.	БИОГРАФИЈА.....	40
9.	ЛИТЕРАТУРА	41

1. УВОД

У овом раду ће бити представљен комплетан циклус једне од примена конволуционих неуронских мрежа. Скупови података који су коришћени у овом раду су скупови података *ibean* [1] и *plantaeK* [2].

Вршено је истраживање у коме су испитане примене различитих архитектура конволуционих неуронских мрежа над скуповима података састављеним од слика. Циљ је да се изгради робустан модел машинског учења који је у стању да разликује болести у биљкама применом аутоматске класификације слика листова у погледу припадности класи здравих или класи болесних. Како би тај циљ био постигнут, било је потребно упознати се са скупом података, припремити га за даљи рад применом различитих техника предобраде података, формирати различите архитектуре модела конволуционих неуронских мрежа, проценити њихове перформансе и коначно, дискутовати о томе који би модел могао бити најбољи и зашто. У листи у наставку налази се кратак опис наредних поглавља у овом раду:

- У поглављу 2 описана је улога вештачког неурона, а затим је назначен значај који се остварује употребом активационе функције. Дат је преглед основне структуре неуронских мрежа и објашњено је основно понашање функције губитка. Дат је опис обучавања итерацијом кроз епохе и представљен је проблем прекомерног прилагођавања.
- У поглављу 3 описан је поступак класификације слика, назначене и у кратким цртама су описане примењиване технологије, као и неопходни поступци који претходе било каквој класификацији, у виду предобраде података. Затим су описане неке од архитектура конволуционих неуронских мрежа и помоћни механизми који се могу користити за обуку модела.
- У поглављу 4 дат је опис скупова података који су коришћени у овом раду, а то су скупови података *ibean* и *plantaeK*. Потом су описани поступци предобраде и учитавања података.
- У поглављу 5 описана је обука свих модела и начини имплементације свих архитектура конволуционих неуронских мрежа.
- У поглављу 6 приказани су резултати добијени на скупу за тестирање и дискутовано је о резултатима.
- У поглављу 7 дат је закључак уз идеје за будући рад над овим скуповима података.
- У поглављу 8 налази се биографија аутора овог рада.
- У поглављу 9 налази се литература коришћена као помоћно средство при изради овог рада

2. КОНВОЛУЦИОНЕ НЕУРОНСКЕ МРЕЖЕ

У овом поглављу описана је основна улога вештачког неурона, дискутовано је о предностима које је могуће остварити употребом активационе функције и напоменуте су неке од најважнијих. Дат је преглед основне структуре неуронских мрежа и напоменута је различита природа понашања која је у вези са функцијом губитка. Дат је опис обучавања итерацијом кроз епохе и представљен је проблем прекомерног прилагођавања.

2.1 ВЕШТАЧКИ НЕУРОН

Вештачка неуронска мрежа је рачунарски модел који ради слично функционисању људског нервних система. Основа сваке вештачке неуронске мреже јесте вештачки неурон [3]. Вештачки неурон описан је скаларном функцијом улаза у облику реалног вектора података, од којих се свака функција дефинисана на тај начин множи одговарајућим тежинским фактором θ који описује допринос улаза при рачунању излаза, при чему излаз вештачког неурона представља један бинарни податак, 0 или 1 [3].

2.2 АКТИВАЦИОНЕ ФУНКЦИЈЕ

Активационе функције се посебно користе у вештачким неуронским мрежама за трансформацију улазног сигнала у излазни сигнал [4]. У вештачкој неуронској мрежи, израчунава се збир производа улаза и њихових одговарајућих тежина и коначно примењује се активациона функција на њега да би се добио излаз тог одређеног слоја и проследио као улаз следећем слоју [4].

Избор активационе функције зависи од проблема који се решава [5]. Познати тип активационе функције јесте Хевисајдова функција, међутим, много чешће се користе непрекидне активационе функције као што су хиперболички тангенс и исправљачка линеарна јединица (енгл. *Rectified Linear Unit – ReLU*), где је за најчешћи избор сигмоидална функција. Претходно наведене функције дефинисане су следећим изразима [2]:

Сигмоидална:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

Хиперболички тангенс:	$\sigma(x) = \frac{e^x - e^{-x}}{e^{-x} + e^{-x}}$	(2)
--------------------------	--	-----

Исправљачка линеарна јединица:	$\sigma(x) = \max(0, x)$	(3)
-----------------------------------	--------------------------	-----

Међутим, неретко се дешава да модуо градијента постане веома мали (проблем нестајућег градијента, енгл. **vanishing gradient**) што доводи до незнатне промене параметара (тежина и помераја) па мрежа престаје да учи [5]. Предност **ReLU** активационе функције у односу на сигмоидалну активациону функцију огледа се у томе што због начина на који је дефинисана за велике вредности улаза не доводи до проблема нестајућег градијента [5].

Још једна активациона функција, која се углавном користи само за излазни слој мреже при решавању класификационих проблема, јесте софтмакс (енгл. **softmax**) функција [5]. Ова функција претвара вектор излазних вредности мреже у расподелу која одређује колика је вероватноћа припадности одговарајућег улаза у мрежу свакој од класа [5].

2.3 СТРУКТУРА НЕУРОНСКИХ МРЕЖА

Први слој неуронске мреже назива се улазним слојем и прима улазне податке (вредности улазних обележја), чиме је дефинисана и његова величина, односно број неурона у њему [5]. Сваки наредни слој као улазе прима излазе из претходног слоја, при чему се за последњи (излазни) слој израчунавају вредности излазних обележја, што одређује и колико неурона он треба да садржи [5]. Слојеви између улазног и излазног називају се скривеним слојевима и могу бити произвољне величине [5]. У овом раду скупови података за обуку садрже слике и одговарајуће класне лабеле свих слика, чиме се за сваки улаз експлицитно дефинише жељени излаз мреже, али не и излази свих скривених слојева [5]. Уколико излаз сваког неурона из одређеног слоја уједно представља улаз сваког неурона наредног слоја, слојеви се сматрају потпуно повезаним (енгл. **fully connected layers**) [5]. Број слојева одређује дубину мреже, због чега се мреже са више слојева и називају дубоким неуронским мрежама (енгл. **deep neural networks – DNN**) [5]. Мреже с којима је могуће дефинисати конволуцију као операцију над улазним подацима, чине конволуционе неуронске мреже (енгл. **convolutional neural networks - CNN**) [5].

2.4 ФУНКЦИЈА ГУБИТКА

Функција губитка, која се такође назива и функција цене, мери компатибилност између предвиђања излаза мреже путем пропагације унапред. [6]. Као и у другим алгоритмима машинског учења, одређивање вредности параметара (у овом случају тежина и померај) врши се у односу на функцију губитка, као меру одступања предвиђених вредности излаза од жељених (циљних) вредности [6].

За минимизацију функције губитка се користе итеративни поступци, углавном засновани на алгоритму опадања градијента. Исход оваквог поступка зависи од иницијалних параметара мреже, тако да се не може гарантовати достизање глобалног минимума [5].

Путања ка минимуму може значајно да осцилује током учења, а различити оптимизациони алгоритми за ажурирање параметара мреже теже да пригуше те осцилације и на тај начин убрзају конвергенцију [5]. Неки од чешће коришћених алгоритама су стохастички алгоритам опадања градијента са моментом (енгл. *stochastic gradient descent with momentum*) и Адам (енгл. *adaptive moment estimation*) [5].

2.5 ОБУЧАВАЊЕ КРОЗ ЕПОХЕ

Једна епоха обуке неуронске мреже представља један пролазак свих узорака за обуку кроз мрежу [5]. Мрежа се типично обучава кроз већи број епоха, обука се прекида када прође унапред одређен број епоха или када се вредност функције цене у унапред одређеном броју узастопних епоха не смањује (или се не мења више од задатог прага) [5]. Вредности параметара мреже могу се ажурирати након читаве епохе, али је то у пракси најчешће неизводљиво због ограничења радне меморије, из тог разлога оне се у пракси најчешће ажурирају након једног проласка једног подскупа узорака задате величине (енг. *batch*) кроз мрежу [5].

Поступак **Dropout** током обуке занемарује излазе појединих неурона са унапред одређеном вероватноћом, изједначавајући их са нулом [5]. На тај начин елиминише међусобну зависност неурона и спречава прекомерно прилагођавање модела [5]. Праћењем грешке на скупу за обуку и валидацију као што је касније описивано у овом раду, може се уочити прекомерно прилагођавање, односно пораст грешке на валидационом скупу и истовремено смањење грешке на скупу за обуку [5]. Обука се у том случају може зауставити пре испуњења унапред постављених критеријума за заустављање, што се назива раним заустављањем (енгл. *early stopping*) [5]. Оно се примењује се када у неколико узастопних епоха не долази до смањења вредности функције губитка на скупу узорака који је унапред издвојен за валидацију [5].

2.6 ПРЕКОМЕРНО ПРИЛАГОЂАВАЊЕ

Проблем прекомерног прилагођавања настаје када мрежа из неког разлога није у стању да учи ефикасно [7]. Ако би модели показали знаке прекомерног прилагођавања, онда бисмо могли да видимо смањену способност да прецизно одредимо генерализоване карактеристике не само за скуп података за обуку, већ и за скупове података за тестирање и валидацију [7]. Што је мање параметара потребних за обуку, мања је вероватноћа да ће се мрежа прекомерног прилагођавања и побољшати предиктивне перформансе модела [7]. Прекомерно прилагођавање је у вези са ситуацијом у којој модел учи статистичке правилности специфичне за скуп за обуку [5]. Ово је један од главних изазова у машинском учењу, пошто модел који је прекомерно прилагођен не може да генерализује податке који никада раније нису били обрађивани. [5]. Нека од решења за прекомерно прилагођавање су [5]:

- повећање података у скупу за обуку,
- повећање укупног броја података,
- регуларизација (опадање тежине, напуштање),
- серијска нормализација и
- смањење сложености архитектуре.

3. КЛАСИФИКАЦИЈА СЛИКА

У овом поглављу је у кратким цртама представљена класификација слика, описане су коришћене технологије, поступци предобrade података, архитектуре конволуционих неуронских мрежа, помоћни механизми за обуку модела, као и мере успешности класификатора коришћене у овом раду.

Технологије коришћене за потребе овог рада јесу програмски језик *Python* и библиотеке *pandas*, *Torch* и *scikit-learn*. Методе предобrade података коришћене у овом раду јесу нормализација, промена величине слике и промена формата слике. Архитектуре конволуционе неуронске мреже које су испитиване у овом раду јесу *LeNet5*, *AlexNet*, *VGG16* и *DenseNet*. За процену перформанси модела на валидационом скупу који је претходно издвојен из скупа за обуку, у овом раду коришћена је обична валидација.

Класификација слика је сложен процес који зависи од различитих фактора. Мере успешности класификатора коришћене у овом раду су тачност (енгл. *accuracy*), прецизност (енгл. *precision*), осетљивост (енгл. *recall*), скор *F1* и уравнотежена тачност (енгл. *balanced accuracy*).

3.1 КОРИШЋЕНЕ ТЕХНОЛОГИЈЕ

Језик *Python* је интерпретирани, објектно оријентисани програмски језик високог нивоа са динамичком семантиком [8]. Синтакса језика *Python* је једноставна, погодна за лако учење, наглашава читљивост и стога смањује трошкове одржавања програма [8]. Језик *Python* подржава различите модуле и пакете, што подстиче модуларност програма и поновну употребу кода [6].

Библиотека *PyTorch* за машинско учење је библиотека отвореног кода (енгл. *open-source*) заснована на програмском језику *Python* и библиотеци *Torch* [9]. Представља једну од платформи за истраживање дубоког учења, а заснована је на апликацијама које користе графичке картице и процесоре [9]. У овом раду, описана библиотека коришћена је за читавање података и у поступку примене *adam* оптимизатора. Такође, она је пружила основ у имплементацији четири различите архитектуре конволуционе неуронске мреже.

Библиотека *Pandas* је библиотека отвореног кода (енгл. *open-source*) која садржи пакете за брзу и ефикасну манипулацију подацима [10].

Библиотека *Scikit-learn* (*sklearn*) једна је од најкоришћенијих библиотека за машинско учење у програмском језику *Python* из тог разлога што пружа избор ефикасних алата за машинско учење и статистичко моделирање укључујући класификацију, регресију, кластеризацију и смањење димензионалности [11]. У овом раду је из библиотеке *sklearn* коришћен модул *sklearn.metrics*, у циљу приказа метрика које указују на квалитет предвиђања модела [12]. Такође,

помоћу модула *sklearn.metrics* генерисане су матрице конфузије које су развијене за класификационе моделе у раду [12]. Матрица конфузије представља одговарајућу табелу која се користи за преглед перформанси класификационог алгорита, тако што визуализује и сумира његове перформансе [13].

3.2 ВАЛИДАЦИЈА

Валидација се дефинише као процес одређивања степена до којег је модел способан да тачно прикаже стваран свет из перспективе употребе за коју је предвиђен [14]. Она се врши након обуке модела, а у комбинацији са обуком поступак валидације има за циљ да лоцира идеалан модел [14]. Валидација представља прогресивно јачање степена поверења за посматрани модел [14].

3.3 МЕРЕ УСПЕШНОСТИ КЛАСИФИКАТОРА

Променљиве које су коришћене за установљавање мера успешности класификатора описане су у наредној листи [15]:

- *TP* (енгл. *true positive*) или број истински позитивних случајева представља исход где модел исправно предвиђа позитивну класу.
- *TN* (енгл. *true negative*) или број истински негативних случајева представља исход где модел исправно предвиђа негативну класу.
- *FP* (енгл. *false positive*) или број погрешно позитивних случајева представља исход где модел нетачно предвиђа позитивну класу.
- *FN* (енгл. *false negative*) или број погрешно негативних случајева представља исход где модел нетачно предвиђа негативну класу.

Кратак опис мера успешности класификатора дат је у листи испод [15]:

- **Тачност** (енгл. *accuracy*) указује на учестаност тачног предвиђања класификатора.

$$Ta\check{c}nost = \frac{TP + TN}{TP + FN + TN + FP} \quad (4)$$

- **Прецизност** (енгл. *precision*) се дефинише као однос тачно класификованих позитивних узорака и укупног броја позитивно класификованих узорака (било исправно или нетачно).

$$Preciznost = \frac{TP}{TP + FP} \quad (5)$$

- **Осетљивост** (енгл. *recall*) представља однос исправно класификованих позитивних узорака од свих узорака који су заправо позитивни.

$$Osetljivost = \frac{TP}{TP + FN} \quad (6)$$

- **Скор *F1*** (енгл. *F1-score*) комбинује прецизност и осетљивост у једну метрику узимајући њихову хармонијску средину.

$$F1 = 2 * \frac{Preciznost * Osetljivost}{Preciznost + Osetljivost} \quad (7)$$

- **Уравнотежена тачност** (енгл. *balanced accuracy*) користи се у бинарним и вишекласним проблемима класификације за решавање проблема неравномерно распоређених скупова података [16]. Дефинише се као просек осетљивости добијен на свакој класи [16].

3.4 АРХИТЕКТУРЕ КОНВОЛУЦИОНЕ НЕУРОНСКЕ МРЕЖЕ

3.4.1 АРХИТЕКТУРА LUNET5

Јан Лекен (енгл. *Yann LeCun*) је са сарадницима предложио су архитектуру неуронске мреже за 1990-их, коју су назвали *LeNet5* [17]. У листи испод дат је опис слојева архитектуре конволуционе неуронске мреже *LeNet5* [18]:

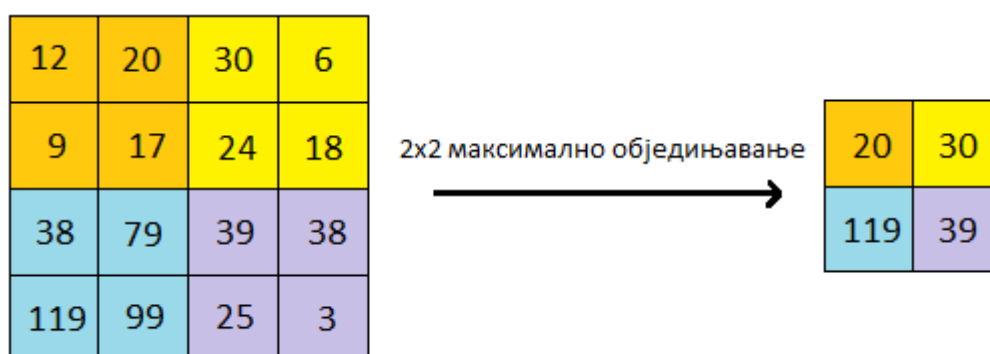
- 1. слој: Улаз у *LeNet5* је слика у нијансама сиве 32×32 пиксела која пролази кроз први конволуциони слој са 6 мапа карактеристика или филтра величине 5×5 и кораком од један. Димензије слике се мењају са $32 \times 32 \times 1$ на $28 \times 28 \times 6$. Након првог конволуционог слоја урађена је дводимензионална нормализација методом *batchNorm2d* која служи за убрзање дубоког мрежног тренинга смањењем унутрашњег померања коваријације [19]. Метода *BatchNorm2d* нормализује активације мреже између слојева у групама [19].
- 2. слој: Следи слој максималног обједињавања (енгл. *max pooling*) са величином филтра 2×2 и кораком два. Добијене димензије слике ће бити смањене на $14 \times 14 \times 6$. На Слици 3.3.1.2 је приказан пример употребе операције максималног обједињавања са величином филтра 2×2 и кораком два.

Максимално обједињавање (енгл. *max pooling*) је операција која бира максимални елемент из региона мапе обележја [20]. Дакле, излаз након слоја максималног обједињавања била би мапа обележја која садржи најистакнутије карактеристике претходне мапе обележја [20]. Операција максималног обједињавања је приказана на Слици 3.4.1.1.

- 3. слој: Постоји други конволуциони слој са 16 мапа обележја величине 5×5 са кораком величине 1. У овом слоју, само 10 од 16 мапа обележја је повезано са 6 мапа обележја претходног слоја. Главни разлог је кршење симетрије у мрежи и одржавање броја веза у разумним границама. Зато је број параметара за обуку у овим слојевима 1516 уместо 2400, број конекција је 151600 уместо 240000.
- 4. слој: Поново следи слој примењује максималног обједињавања са величином филтра 2×2 и кораком од 2. Овај слој је исти као други слој

осим што има 16 мапа карактеристика тако да ће излаз бити смањен на $5 \times 5 \times 16$.

- 5. слој: Овај слој је потпуно повезан конволуциони слој са 120 мапа обележја од којих је свака величине 1×1 . Свака од 120 мапа обележја у петом слоју је повезана са свих 400 чворова ($5 \times 5 \times 16$) у четвртом слоју.
- 6. слој: Представља слој који је потпуно повезан слој са 84 мапе обележја.
- 7. слој: Излазни слој је потпуно повезани *softmax* слој са 3 могуће вредности које одговарају цифрама од 0 до 2 у зависности од тога којој класи слика припада.



Слика 3.4.1.1 – Пример употребе операције максималног обједињавања

Као што је раније напоменуто, улаз архитектуре *LeNet5* је слика у нијансама сиве, величине 32×32 пиксела, из тог разлога, пре него што слике буду убачене у мрежу, потребно им је променити димензије са $500 \times 500 \times 3$ пиксела на $32 \times 32 \times 3$ пиксела. У скуповима података *ibean* и *plantaeK* налазе се слике у боји па је повратна вредност димензионалности таквих слика једнака 3. Уместо стандардне бинарне слике која се генерално користи у архитектури *LeNet5*, за улаз је коришћена слика у боји, димензија $32 \times 32 \times 3$ пиксела.

3.4.2 АРХИТЕКТУРА ALEXNET

Архитектуру конволуционе неуронске мреже *AlexNet* је дизајнирао Алекс Крижевски (енгл. *Alex Krizhevsky*) са сарадницима [21]. Архитектура конволуционе неуронске мреже *AlexNet* је дубока 16 слојева [21]. Њен улаз је слика величине 227 пиксела у боји која пролази кроз први конволуциони слој [21]. У листи испод дат је опис слојева архитектуре конволуционе неуронске мреже *AlexNet* [22]:

- 1. слој: Конволуција са 96 језгара (енгл. *kernels*) величине $11 \times 11 \times 3$ и корак од четири пиксела, након чега следи максимално обједињавање (енгл. *max pooling*) величине 3×3 са кораком величине два. Димензије излаза су $55 \times 55 \times 96$ након првог конволуционог слоја, и $27 \times 27 \times 96$ након максималног обједињавања.
- 2. слој: Конволуција са 256 језгара величине $5 \times 5 \times 96$ и кораком један, праћено слојем максималног обједињавања величине 3×3 и кораком два. Димензије излаза су $27 \times 27 \times 256$ након другог слоја и $13 \times 13 \times 256$ после максималног обједињавања.
- 3. слој: Конволуција са 384 језгара величине $3 \times 3 \times 256$, са кораком величине један, без обједињавања. Димензије излаза су $13 \times 13 \times 384$ након трећег слоја.
- 4. слој: Конволуција са 384 језгара величине $3 \times 3 \times 384$, са кораком од један, без обједињавања. Димензије излаза су $13 \times 13 \times 384$ након четвртог конволуционог слоја.
- 5. слој: Конволуција са 256 језгара величине $3 \times 3 \times 384$ и кораком од један, праћено слојем максималног обједињавања величине 3×3 и кораком од два. Димензије излаза су $13 \times 13 \times 256$ након петог конволуционог слоја и $6 \times 6 \times 256$ након максималног обједињавања.
- 6. и 7. слој: Потпуно повезани слојеви, где и шести и седми слој имају 4096 неурона.
- 8. конволуциони слој: Последњи слој је *softmax* слој.

3.4.3 АРХИТЕКТУРА VGG16

Архитектура конволуционе неуронске мреже **VGG16** је дубока 16 слојева [23]. Први конволуциони слој прима слику величине 224×224 пиксела као улаз. Корак конволуције се узима као један пиксел у свим слојевима. Сви слојеви са максималним обједињавањем су прозора величине 2×2 са кораком два [23]. У листи испод дат је опис слојева архитектуре конволуционе неуронске мреже **VGG16** [24]:

- 1. и 2. слој: Конволуција са 64 језгара у првом слоју величине $3 \times 3 \times 3$, док је у другом величина језгра $3 \times 3 \times 64$, након чега следи слој максималног обједињавања.
- 3. и 4. слој: Конволуција са 128 језгара у трећем слоју величине $3 \times 3 \times 64$, док је у четвртом величина језгра $3 \times 3 \times 128$, након чега следи слој максималног обједињавања.
- 5, 6 и 7. слој: Конволуција са 256 језгара величина је редом $3 \times 3 \times 128$, $3 \times 3 \times 256$, $3 \times 3 \times 256$, након чега следи слој максималног обједињавања.
- 8, 9 и 10. слој: Конволуција са 512 језгара величина је редом $3 \times 3 \times 256$, $3 \times 3 \times 512$, $3 \times 3 \times 512$, након чега следи слој максималног обједињавања.
- 11, 12 и 13. слој: Конволуција са 512 језгара величина је редом $3 \times 3 \times 512$, $3 \times 3 \times 512$, $3 \times 3 \times 512$, након чега следи слој максималног обједињавања.
- 14, 15 и 16. слој: Након скупа конволуционих слојева, постоје три потпуно повезана слоја са редом 4096, 4096 и 1000 неурона. У потпуно повезаном слоју, сваки неурон прихвата улаз од активација неурона претходног слоја. Последњи слој је *softmax* слој.

3.4.4 АРХИТЕКТУРА *DENSENET*

Улаз архитектуре *DenseNet* је слика у боји величине $150 \times 150 \times 3$ пиксела, која пролази кроз први конволуциони слој [25]. Ова архитектура се састоји од серије конволуционих слојева, а сваки од њих прати слој за активацију исправљене линеарне јединице (***ReLU***), који се на крају спаја са потпуно повезаном неуронском мрежом са ***softmax*** бинарном класификатором [25]. Сваки конволуциони слој има величину филтра 3×3 без додавања и корака 1×1 (осим за први слој који има величину филтра 5×5), генеришући 128 мапа карактеристика [25]. У складу са бројем конволуционих слојева, њиховом величином кернела, минимална величина улаза од 150×150 је обезбеђена стварајући коначни улаз од $150 \times 150 \times 3$ [25]. Након пет конволуционих слојева са ***ReLU*** активационом функцијом додаје се слој за како би се генерисале векторске карактеристике које ће се увести у потпуно повезану неуронску мрежу, која се састоји од четири слоја: три слоја са по 2000 неурона и ***ReLU*** активационом функцијом [25]. На крају се налази слој са два неурона који користи ***softmax*** активациону функцију [25].

4. СКУПОВИ ПОДАТАКА

У овом поглављу биће представљени скупови података *ibean* и *plantaeK*, као и процес предобраде података, који се састоји од три трансформације. Све три трансформације биће извршене при учитавању података који ће претворити податке у облик погодан за касније фазе обучавања мреже.

4.1 СКУП ПОДАТАКА *IBEAN*

Скуп података *ibean* представља слике лишћа које је на терену у различитим окрузима у Уганди направила лабораторија под називом *Makerere AI lab*, у сарадњи са Националним институтом за истраживање ресурса усева (енгл. *NaCRRI*), националним телом задуженим за пољопривредна истраживања у Уганди [1].

Подаци укључују слике листова који су детектовани на терену [1]. Сlike су првобитно снимљене помоћу паметног телефона, а затим су анотирани од стране стручњака запослених у Националном институту за истраживања ресурса усева, који су за сваку слику одредили која се болест манифестује [1]. Експерти су чинили део тима за прикупљање података, а слике су биле означене директно током процеса прикупљања података на терену [1]. Скуп података *ibean* је доступан преко репозиторијума код којег је назначена лиценца *MIT* (<https://opensource.org/licenses/MIT>).

Подаци скупа података *ibean* се односе на слике листова које су представљене у три класе [1]:

- класа *Здрави листови*,
- класа *Листови са рђом* и
- класа *Листови са бољешћу лисне пеге*.

Скуп података под називом *ibean* садржи укупно 1296 узорака, распоређених на следећи начин [1]:

- класа *Здрави листови* броји 428 узорака,
- класа *Листови са рђом* броји 432 узорака и
- класа *Листови са бољешћу лисне пеге* броји преосталих 436 узорака.

На основу приложених података, приметно је да је број узорака из сваке класе приближно једнак, стога у овом случају неће бити проблема што се тиче неравномерне расподеле података између класа [1]. Скуп података *ibean* је адекватно подељен на скупове за тренирање, тестирање и валидацију [1].

Скуп за обуку који служи за формирање класификатора састоји се од 1035 узорака распоређених на следећи начин [1]:

- 342 узорака из класе *Здравих листова пасуља*,
- 348 узорака из класе *Листова са рђом* и
- 345 узорака из класе *Листова са бољешћу лисне пеге*.

Скуп за тестирање који служи за процену перформанси код обученог класификатора састоји се од 128 узорака од којих је [1]:

- 42 узорака из класе *Здравих листова пасуља*,
- 43 узорака из класе *Листова са рђом* и
- 43 узорака из класе *Листова са бољешћу лисне пеге*.

Осим скупова за обуку и за тестирање, у скупу података *ibean* налази се и скуп за валидацију који служи за подешавање слободних параметара класификатора [1]. У скупу за валидацију се налази укупно 133 узорка од којих је [1]:

- 44 узорака из класе *Здравих листова пасуља*,
- 45 узорака из класе *Листова са рђом* и
- 44 узорака из класе *Листова са бољешћу лисне пеге*.

Свака слика из скупа *ibean* је димензије 500×500 пиксела [1]. С обзиром на то да је димензионалност слика у боји једнака три, коначна димензија слика износи 500×500×3 пиксела [1]. Свака слика заузима од 100 до 200 килобајта меморије [1]. На сликама је у првом плану приказан лист, док позадину сваке слике чине земља, камење, листови, неки други делови биљака, као и делови шаке [1].

4.2 СКУП ПОДАТАКА *PLANTAEK*

Скуп података *plantaeK* представља скуп слика лишћа које су снимили пољопривредници на дневном светлу професионалним дигиталним апаратом у индијском региону Кашмир [2]. Скуп података *plantaeK* дат је са лиценцом *CC BY 4.0* (<https://creativecommons.org/licenses/by/4.0/>). Скуп података су поставили Коур и Арора са Универзитета Шри Мата Ваишно Деви [2]. У скупу података *plantaeK* налазе се слике листова јабуке, кајсије, трешње, бруснице, грожђа, брескве, крушке и ораха које су распоређене у различите директоријуме [2]. Један директоријум садржи једну врсту листа [2]. За сваку биљку бирају се две врсте листова, једна здрава, а друга болесна [2].

Сваки од директоријума скупа података *plantaeK* састоји се од два поддиректоријума [2]:

- *здрави листови* и
- *болесни листови*

Све слике су из свих поддиректоријума здравих листова пребачене у један директоријум који садржи здраве листове јабуке, кајсије, трешње, бруснице, грожђа, брескве, крушке и ораха. Исти поступак је урађен за све поддиректоријуме болесних листова.

Сада се директоријум здравих листова састоји од 1219 слика, док се директоријум болесних листова састоји од 934 листова различитих врста биљака [2]. Из овог броја узорака видимо да је број узорака из сваке класе приближно једнак. Сходно томе, очекивано је да неће настати проблем који је у вези са неравномерном расподелом података између класа.

Свака слика из скупа *plantaeK* је димензије 6000×4000 пиксела [2]. Сlike су дате у боји, стога коначна димензија слика износи 6000×4000×3 [2]. Свака од слика заузима од 1,6 до 1,88 мегабајта меморије [2]. На сликама је у првом плану приказан лист, док је позадина сваке слике сива подлога.

4.3 ПРЕДОБРАДА ПОДАТАКА

За потребе даљег истраживања и креирања модела рађена је предобрада слика. За почетак извршена је трансформација слика методом *transforms.Compose* која садржи неколико метода [26].

Као први аргумент у позиву методе *transforms.Compose* коришћен је позив методе *transforms.Resize* којом је одрађена промена величине слике [27]. Овом методом смањена је величина слике у зависности од улаза архитектуре конволуционе неуронске мреже [27]. Величина слике је смањена са димензија 500×500×3 на димензије 32×32×3, 227×227×3, 224×224×3 и 150×150×3 за редом архитектуре *LeNet5*, *AlexNet*, *VGG16* и *DenseNet*.

Следећа трансформација, као други аргумент у позиву методе *transforms.Compose* коришћен је позив методе *transforms.ToTensor*, која конвертује слику типа *PIL Image* димензија 150×150×3 са пиксел доменом [0, 255] на слику типа *torch.FloatTensor* димензија 3×150×150 у домену [0.0, 1.0] [28].

Осим промена величине слике и конвертовања исте, одрађена је и нормализација трећим аргументом у позиву методе *transforms.Compose*. За нормализацију слика коришћена је позив методе *transforms.Normalize* ([0.5, 0.5, 0.5], [0.5, 0.5, 0.5]) [29]. Параметри методе *transforms.Normalize* представљају редом [29]:

1. низ средњих вредности и
2. низ стандардних девијација.

Овом методом се мења домен са [0.0, 1.0] на [-1.0, 1.0] тако да нови пиксели буду рачунати по формули [26] :

$$Izlaz = Ulaz - \frac{Srednja\ vrednost}{Standardna\ devijacija} \quad (8)$$

Методом нормализације завршена је предобрада података.

4.4 УЧИТАВАЊЕ ПОДАТАКА

За учитавање података коришћена је из библиотеке *Torch* класа *DataLoader* помоћу које је могуће креирати или учитати скуп података [30]. Кроз овај скуп података моћи ће да се врши итерација [30]. Са овом класом могуће је још прилагођавање редоследа учитавања података, аутоматско дозирање (енгл. *Batching*), једно- и вишепроцесно учитавање података, као и аутоматско заузимање меморије [30].

Први аргумент у позиву конструктора класе *DataLoader* је позив методе метода *torchvision.datasets.ImageFolder* која има два параметра [31]. *Torchvision.datasets.ImageFolder* представља путању директоријума у којем се налазе слике, док њен други параметар представља трансформацију слика која је дефинисана у поглављу 4.3 за предобраду података [31].

Као други аргумент у позиву конструктора класе *DataLoader* налази се величина серије (енгл. *batch size*) који показује колики ће број узорака бити пропагиран кроз мрежу [32]. Вредност величине серије као другог параметра у класи *DataLoader* зависи од архитектуре неуронске мреже која се користи [29]. Вредност величине серије је редом 8, 16, 32, 64 за архитектуру конволуционе неуронске мреже *LeNet5*, *AlexNet*, *VGG16* и *DenseNet*.

5. ОБУКА МОДЕЛА

За све коришћене архитектуре *LeNet5*, *AlexNet*, *VGG16* и *DenseNet* за први скуп података *ibean*, налази се:

- 1035 узорака за *обуку*,
- 133 узорка за *валидацију* и
- 128 узорака за *тестирање*.

За други скуп података *planteK*, такође за горе наведене архитектуре, првобитно је извршена подела на тренинг скуп и на скуп који ће бити коришћен за валидацију и тестирање. Подела је извршена тако да тренинг скупу података припадне 75%, а тест и валидационом скупу података 25% од укупног броја узорака, при чему је вођено рачуна да расподела инстанци по класама остане у истом односу. Након прве, уследила је и друга подела тако што је валидациони и тест скуп подељен. Од 25% укупног броја узорака 50% је припало валидационом скупу, док је 50% припало тест скупу. Тако да се за све коришћене архитектуре овом раду налази:

- 1609 узорака за *обуку*,
- 268 узорака за *валидацију* и
- 269 узорака за *тестирање*.

Вредности параметара за сваку од поменутих архитектура одређиване су комбиновањем различитих вредности и обучавањем више модела користећи валидацију. Мера успешности класификатора која је коришћена за одабир оптималних вредности параметара је функција губитка модела над валидационом и тренинг скупу. Коначне перформансе модела добијене су на скупу за тестирање.

Компајлирање модела је потребно да би се модел финализирао и учинио потпуно спремним за употребу. За компајлирање, мора се навести оптимизатор, функција губитка и метрика за оцену перформанси. За сваки модел коришћен је оптимизатор *adam*, док је функција губитка *categorical_crossentropy*. Као метрика је посматрана класификација тачности. Параметар *epochs* представља број епоха обуке, и за вредност овог параметра коришћене су различите вредности у односу на архитектуру која је употребљена за креирање модела [33].

У овом раду све разматране архитектуре конволуционе неуронске мреже су имплементиране помоћу библиотеке *Torch*. У овом поглављу биће представљене имплементације слојева који су коришћени при обуци и одабир параметара модела, док су значења самих слојева описана у поглављу 3.4.

5.1 МОДЕЛ СА АРХИТЕКТУРОМ *LENET5*

Имплементација модела са архитектуром конволуционе неуронске *LeNet5* у првом и другом слоју претежно се ослања на употребу четири методе, а то су:

- ***Conv2d*** (*int, int, kernel_size, stride, padding*) – примењује 2D конволуцију на улазни сигнал [34]
 - први параметар описује број улазних канала;
 - други параметар описује број излазних канала;
 - трећи параметар ***kernel_size*** описује величину кернела за примену конволуције;
 - четврти параметар ***stride*** означава примењивани корак;
 - пети параметар ***padding*** представља проширење над полазном сликом, у циљу очувања димензија полазне слике након примене конволуције.
- ***BatchNorm2d*** (*int*) – примењује нормализацију серије на 4D улаз [19];
- ***ReLU()*** – представља активациону функцију ***ReLU*** [35];
- ***MaxPool2d*** (*kernel_size, stride*) – примењује 2D максимално обједињавање на улазни сигнал [36]
 - први параметар ***kernel_size*** описује величину филтра конволуције;
 - други параметар ***stride*** означава примењивани корак.

Вредности параметара у методи ***Conv2d*** кроз слојеве приказани су у Табели 5.1.1.

	Вредност 1. параметра	Вредност 2. параметра	Вредност 3. параметра	Вредност 4. параметра	Вредност 5. параметра
1. слој	3	6	5	1	0
2. слој	6	16	5	1	0

Табела 5.1.1 – Вредности параметра у методи *Conv2d* кроз прва два слоја

Вредности параметра методе ***BatchNorm2d*** за први и други слој су редом шест и шеснаест. Трећи, пети и седми слој примењује линеарну трансформацију која је имплементирана методом ***Linear*** код које параметри означавају:

- први параметар означава број улазних и
- други параметар означава број излазних канала.

Вредности параметара у методи ***Linear*** кроз слојеве приказани су у Табели 5.1.2.

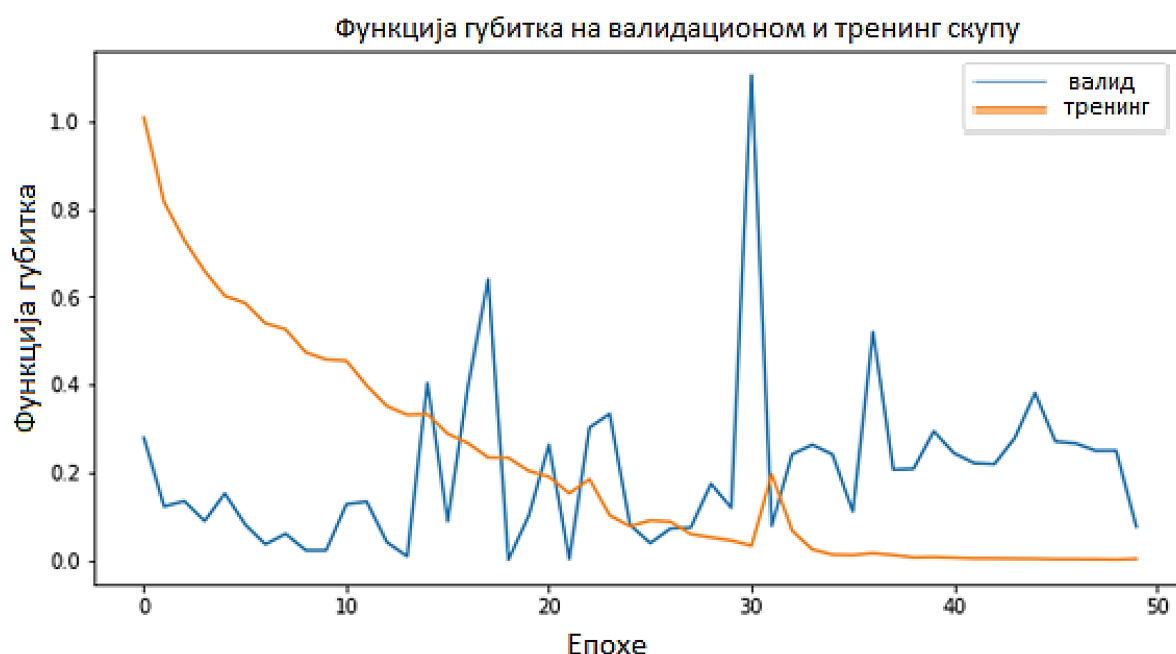
	Вредност 1. параметра	Вредност 2. параметра
3. слој	400	120
5. слој	120	84
7. слој	84	број класа

Табела 5.1.2 – Вредности параметра у методи *Linear* у 3, 5 и 7. слоју

Након трећег и петог слоја имплементирани су слојеви у којима се користи активациона функција **ReLU** која је подржана методом **ReLU**.

5.1.1 ОБУКА ЗА СКУП ПОДАТАКА *IBEAN*

Слика 5.1.1.1 представља промену функције губитка модела кроз епохе, при чему крива плаве боје представља одговарајућу промену над валидационим скупом података, а крива наранџасте боје представља дату промену над скупом података намењеним за тренинг. Поступак обуке модела обухватио је 50 епоха. На горепоменутој слици могуће је уочити и да функција губитка модела на валидационом скупу података достиже нагли пораст након 14. епохе. Вредности које таква функција има на тренинг и валидационом скупу су редом 0,3318 и 0,0016. За број епоха изабрана је вредност 14.



Слика 5.1.1.1 – Функција губитка модела са архитектуром *LeNet5*, посебно на тренинг и валидационом скупу за скуп података *ibean*

5.1.2 ОБУКА ЗА СКУП ПОДАТАКА *PLANTAEK*

Обука модела са архитектуром конволуционе неуронске мреже *LeNet5* за скуп података *plantaeK* извршена је у 50 епоха. Функција губитка модела над валидационим скупом података бележи постепени пад до 16. епохе, када и достиже своју минималну вредност. Са друге стране, функција губитка модела над скупом података намењеним за тест константно опада до 27. епохе, а након тога њена природа постаје променљива. За вредност броја епоха за обуку коришћена је вредност 16. У описаном случају, вредности

функције губитка модела које су забележене на валидационом и тест скупу, након шеснаесте епохе, једнаке су 0,2932 и 0,0702, редом.

5.2 МОДЕЛ СА АРХИТЕКТУРОМ *ALEXNET*

Имплементација модела са архитектуром *AlexNet* се састоји од осам слојева, где се првих пет претежно ослања на употребу истих метода чије је значење описано у одељку 5.1, а то су:

- ***Conv2d*** (*int, int, kernel_size, stride, padding*),
- ***BatchNorm2d*** (*int*) и
- ***ReLU***().

Вредности параметара у методи ***Conv2d*** кроз слојеве приказани су у Табели 5.2.1.

	Вредност 1. параметра	Вредност 2. параметра	Вредност 3. параметра	Вредност 4. параметра	Вредност 5. параметра
1. слој	3	96	11	4	0
2. слој	96	256	5	1	2
3. слој	256	384	3	1	1
4. слој	384	384	3	1	1
5. слој	384	256	3	1	1

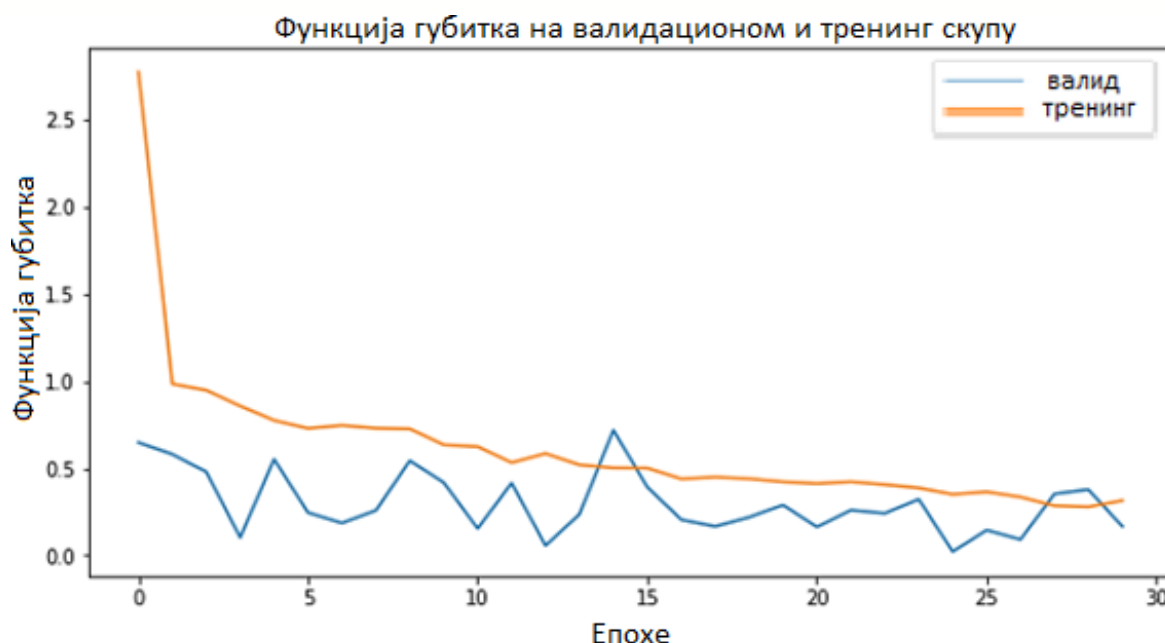
Табела 5.2.1 – Вредности параметра у методи *Conv2d* кроз првих пет слојева

Први, други и пети слој се разликују по томе што након метода ***Conv2d***, ***BatchNorm2d*** и ***ReLU*** примењује максимално обједињавање које је имплементирано методом ***MaxPool2d*** где за вредности параметара *kernel_size* и *stride* имају редом вредности три и два.

Следе шести и седми слој који садрже методу ***Dropout*** са једним параметром чија је вредност једнака 0,5. Осим методе ***Dropout*** у шестом и седмом слоју примењена је линеарна трансформација која је имплементирана методом ***Linear***. Вредности параметара методе ***Linear*** у шестом слоју су редом 9216 и 4096, док су у седмом слоју вредности 4096 и 4096. Последња метода у овим слојевима је ***ReLU***. Последњи слој садржи само методу ***Linear*** са вредностима 4096 и број класа, која зависи од скупа података над којим се обучава ова неуронска мрежа.

5.2.1 ОБУКА ЗА СКУП ПОДАТАКА *IBEAN*

Слика 5.2.1.1 приказује промену функције губитка модела кроз епохе, при том користећи аналогне ознаке оним описаним у одељку 5.1.2 за представу кривих – плава боја у вези је са валидационим скупом података, а наранаста боја представља промене над скупом података који је намењен за тест. Поступак обуке модела заустављен је након 30 епоха. На слици која је доле приказана могуће је уочити да функција губитка модела константно опада на тренинг скупу података, док на валидационом скупу података показује променљиво понашање. Из разлога што се након четврте епохе уочавају интензивне, неруранвотежене промене код функцији губитка на валидационом и тренинг скупи, за вредност броја епоха за обуку усвојена је вредност четири. Вредности функције губитка модела на валидационом и тест скупу, након четврте епохе износе редом 0,8585 и 0,1039.



Слика 5.2.1.1 – Функција губитка модела са архитектуром *AlexNet*, посебно на тренинг и валидационом скупу за скуп података *ibean*

5.2.2 ОБУКА ЗА СКУП ПОДАТАКА *PLANTAEK*

Обука модела са архитектуром конволуционе неуронске мреже *AlexNet* за скуп података *plantaeK* извршена је у 30 епоха. Функција губитка модела над валидационим скупом података бележи константни пад све до последње посматране епохе. Са друге стране, функција губитка модела на тест скупу података константно опада до једанаесте епохе, а то је такође и редни број епохе почевши од које описана функција почиње са постепеним растом. За вредност броја епоха за обуку усвојена је вредност једанаест. Вредности које

достиге функција губитка модела на валидационом и тест скупу након једанаесте епохе једнаке су 0,3474 и 0,0978, редом.

5.3 МОДЕЛ СА АРХИТЕКТУРОМ VGG16

Имплементација модела са архитектуром **VGG16** састоји се шеснаест слојева, где се првих тринаест претежно ослања на употребу три методе, а то су:

- **Conv2d** (*int, int, kernel_size, stride, padding*),
- **BatchNorm2d** (*int*) и
- **ReLU**().

Други, четврти, седми, десети и тринаести слој се разликују по томе што након ових примењује максимално обједињавање које је имплементирано методом **MaxPool2d** где за вредности параметара **kernel_size** и **stride** имају вредности редом два и два.

Вредности првог и другог параметара у методи **Conv2d** кроз слојеве приказани су у Табели 5.3.1.

	Први параметар методе Conv2d : величина улаза канала	Други параметар методе Conv2d : величина излаза канала
1. слој	3	64
2. слој	64	64
3. слој	64	128
4. слој	128	128
5. слој	128	256
6. слој	256	256
7. слој	256	256
8. слој	256	512
9. слој	512	512
10. слој	512	512
11. слој	512	512
12. слој	512	512
13. слој	512	512

Табела 5.3.1 – Вредности првог и другог параметра у методи **Conv2d** кроз 13 слојева

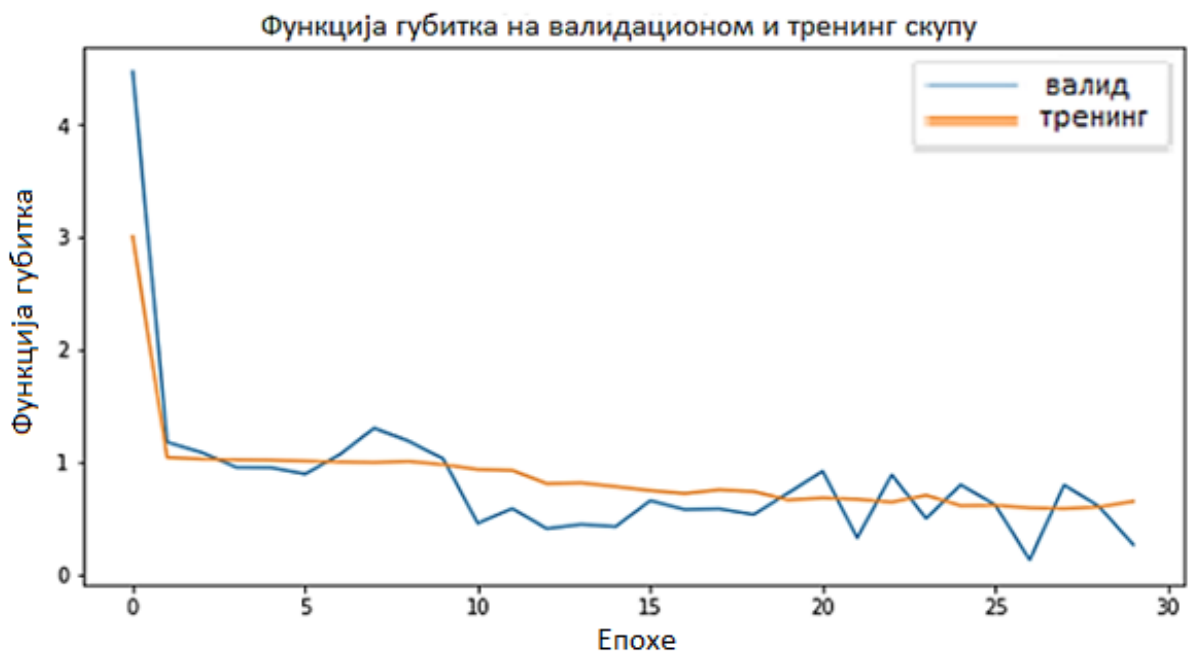
Вредности параметара **kernel_size, stride, padding** у методи **Conv2d** у тринаест слојева износе редом три, један и један. Четрнаести и петнаести слој се претежно ослања на употребу три методе редом **Dropout, Linear** и **Relu**. Ова два слоја се једино разликују у вредностима параметара код методе **Linear**. Четрнаести слој у методи **Linear** има вредности 8192 и 4096, док петнаести слој има вредности 4096 и 4096. Последњи слој садржи само методу **Linear** са

вредностима параметара 4096 и број класа, који зависи од скупа података коришћеног за обучавање неуронске мреже.

5.3.1 ОБУКА ЗА СКУП ПОДАТАКА IBEAN

Аналогно поменутиим моделима у одељцима 5.2.1 и 5.2.2, над скупом података *ibean*, крива плаве боје представља промену функције губитка модела која је остварена на валидационом скупу кроз епохе, док крива обојена наранџастом означава промену функције губитка модела која се односи на скуп података намењен за тестирање.

Обука модела извршена је у 30 епоха. Слика 5.3.1.1 приказује благи пад функције губитка модела кроз епохе на скупу података намењеном за тест, док су вредности ове функције на валидационом скупу података забележене као променљиве. За вредност броја епоха за обуку усвојена је вредност 27. Вредности функције губитка модела на валидационом и тест скупу након 27. епохе износе редом 0,5942 и 0,1310.



Слика 5.3.1.1 – Функција губитка модела са архитектуром VGG16, посебно на тренинг и валидационом скупу за скуп података *ibean*

5.3.2 ОБУКА ЗА СКУП ПОДАТАКА *PLANTAEK*

Обука модела са архитектуром конволуционе неуронске мреже *VGG16* за скуп података *plantaeK* извршена је у 30 епоха. Овај случај бележи константан пад функције губитка модела након треће епохе, како на валидационом, тако и на тренинг скупу. Као број епоха за обуку усвојена је вредност четрнаест. Вредности функције губитка модела, након четрнаесте епохе, на валидационом и тест скупу, једнаке су редом 0,3949 и 0,0840.

5.4 МОДЕЛ СА АРХИТЕКТУРОМ *DENSENET*

Имплементација модела са архитектуром *DenseNet* претежно се ослања на употребу слојева који су директно међусобно повезани користећи блокове под називом *DenseBlocks*. Класу *DenseNet* описују параметри:

- ***growth_rate* (int)**: број филтара који се додаје сваком слоју
- ***block_config*** (листа од 4 *int*): број слојева у сваком блоку за обједињавање
- ***num_init_features* (int)**: број филтара који је потребно научити у сваком конволуционом слоју
- ***bn_size* (int)**: мултипликативни фактор
- ***drop_rate* (float)**: *dropout* оцена након сваког *dense* слоја
- ***num_classes* (int)**: број класа за класификацију
- ***memory_efficient* (bool)**: опис мере ефикасности

Подразумеване вредности које параметри ове класе имају су: ***growth_rate* = 32**, ***block_config* = (6, 12, 24, 16)**, ***num_init_features* = 64**, ***bn_size* = 4**, ***drop_rate* = 0**, ***num_classes* = 1000**, ***memory_efficient* = False**.

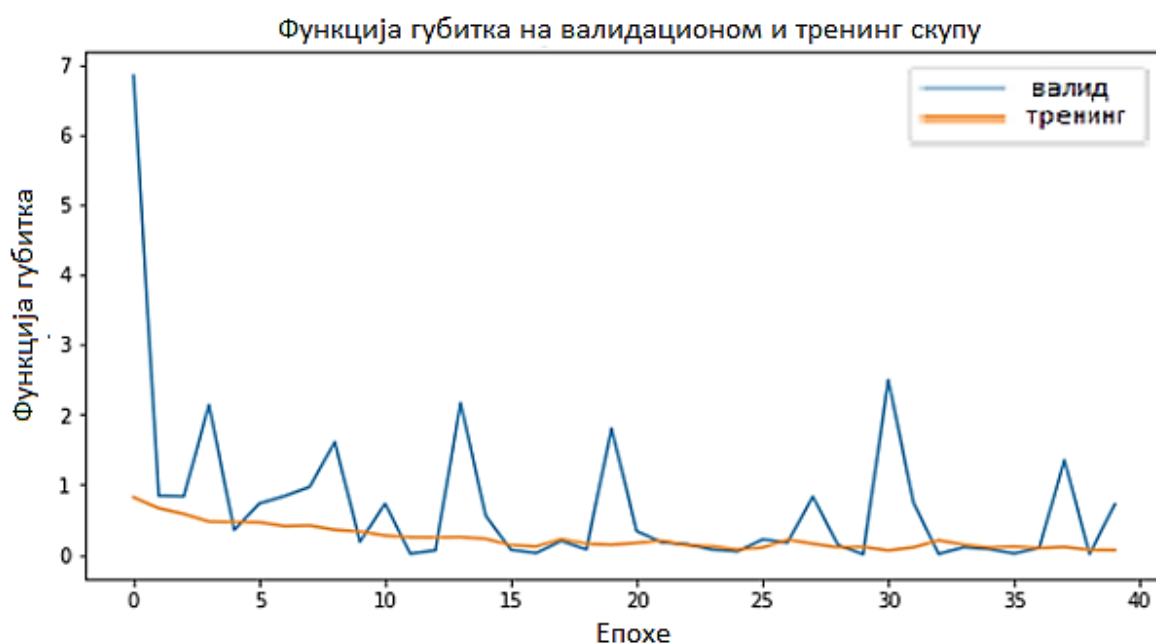
У Табели 5.4.1 која следи наведена су четири различита модела заснована на повезаним конволуционим неуронским мрежама које се базирају на претходно описаној класи *DenseNet*.

модел	<i>growth_rate</i>	<i>block_config</i>	<i>num_init_features</i>	<i>pretrained</i>	<i>progress</i>
densenet121	32	(6, 12, 24, 16)	64	False	True
densenet161	48	(6, 12, 36, 24)	96	False	True
densenet169	32	(6, 12, 32, 32)	64	False	True
densenet201	32	(6, 12, 48, 32)	64	False	True

Табела 5.4.1 – Вредности параметара за четири различита модела

5.4.1 ОБУКА ЗА СКУП ПОДАТАКА *IBEAN*

Слика 5.4.1.1 представља промену функције губитка модела кроз епохе, при чему крива плаве боје представља одговарајућу промену над валидационим скупом података, а крива наранџасте боје представља дату промену над скупом података намењеним за тренинг. Поступак обуке модела обухватио је 40 епоха. На приложеној слици могуће је уочити да функција губитка модела на тренинг скупу података бележи постепени пад почевши од треће епохе, док су вредности функције губитка модела на валидационом скупу података променљиве. Вредности које таква функција има на тренинг и валидационом скупу су редом 0,0125 и 0,9256. За број епоха изабрана је вредност три.



Слика 5.4.1.1 – Функција губитка модела са архитектуром *DenseNet*, посебно на тренинг и валидационом скупу за скуп података *ibean*

5.4.2 ОБУКА ЗА СКУП ПОДАТАКА *PLANTAEK*

Обука модела са архитектуром конволуционе неуронске мреже *DenseNet* за скуп података *plantaeK* извршена је у 40 епоха. Овај случај бележи константан пад функције губитка модела након осме епохе на валидационом скупу података, док је на тренинг скупу забележен нагли пад почевши од пете епохе. Као број епоха за обуку усвојена је вредност осам. Вредности функције губитка модела, након осме епохе, на валидационом и тест скупу, једнаке су редом 0,4213 и 0,0247.

6. РЕЗУЛТАТИ И ДИСКУСИЈА

У овом поглављу ће бити приказане перформансе сваког од модела добијене на скупу за тестирање, биће дискутовано о времену које је потребно за обуку модела. Такође, биће дискутовано о томе који би модел архитектуре конволуционе неуронске мреже могао бити најбољи и зашто.

6.1 ПЕРФОРМАНСЕ МОДЕЛА

6.1.1 ПЕРФОРМАНСЕ НА СКУПУ ПОДАТАКА *IBEAN*

С обзиром на то да се у овом раду за скуп података *ibean* решава вишекласни класификациони проблем, за мере прецизности, осетљивости и скор *F1* биће узет у обзир просек за сваку класу. Када се осетљивост израчуна на овакав начин, добије се исти резултат као и за тачност те ће при евалуацији перформанси бити приказана само тачност.

У Табели 6.1.1.1 приказане су коначне перформансе модела добијене на скупу за тестирање. За сваки модел дате су перформансе добијене коришћењем вредности параметара које су дале најбоље резултате на скупу за обучавање.

Архитектура	Тачност (%)	Прецизност (%)	Скор <i>F1</i> (%)
LeNet5	80,47	81,19	80,60
AlexNet	82,03	81,90	81,92
VGG16	89,06	89,54	89,12
DenseNet	88,28	88,34	88,29

Табеле 6.1.1.1 – Перформансе модела добијене на скупу за тестирање за скуп података *ibean*

При поређењу перформанси сваког од модела из Табеле 6.1.1.1, може бити уочено да за исту архитектуру конволуционе неуронске мреже тачност, прецизност, скор *F1* и осетљивост не одступају много једна од друге.

За све три мере успешности, модел са архитектуром **VGG16** даје најбоље перформансе, док модел са архитектуром **LeNet5** даје убедљиво најлошије перформансе по свакој од три мере успешности. Ни у једном случају се не дешава ситуација да је неки модел бољи од неког другог по једној мери успешности, а други бољи по некој другој мери успешности тако да може да се подвуче јасна црта који модели дају најбоље резултате. Модели са архитектурама конволуционе неуронске мреже **DenseNet** и **VGG16** су се показали најбоље по питању перформанси при чему је максимална разлика у њиховим међусобним перформансама за било коју меру успешности 1,2%.

Најлошије су се показали модели чије архитектуре спадају у класичне мрежне архитектуре (*LeNet5* и *AlexNet*), при чему треба напоменути да је модел са архитектуром *AlexNet* имао боље перформансе него модел са архитектуром *LeNet5*.

6.1.2 ПЕРФОРМАНСЕ НА СКУПУ ПОДАТАКА *PLANTAEK*

За разлику од скупа података *ibean* где се решава вишекласни проблем класификације, за скуп података *plantaeK* се решава бинарни класификациони проблем. Као и код скупа података *ibean*, израчунавањем осетљивости добија се исти резултат као и за тачност па ће при евалуацији перформанси бити приказани само тачност, прецизност и скор *F1*.

У Табели 6.1.2.1 приказане су коначне перформансе модела добијене на скупу за тестирање. За сваки модел дате су перформансе добијене коришћењем вредности параметара које су дале најбоље резултате на скупу за обучавање за скуп података *plantaeK*.

Архитектура	Тачност (%)	Прецизност (%)	Скор <i>F1</i> (%)
LeNet5	75,37	75,29	75,79
AlexNet	76,43	76,88	76,67
VGG16	80,06	80,98	80,67
DenseNet	82,28	82,34	82,29

Табела 6.1.2.1 – Перформансе модела добијене на скупу за тестирање за скуп података *plantaeK*

При поређењу перформанси сваког од модела из Табеле 6.1.2.1, може бити уочено, као и код скупа података *ibean*, да за исту архитектуру конволуционе неуронске мреже вредности мера успешности не одступају много једна од друге.

За све три мере успешности, у овом случају за скуп података *plantaeK*, модел са архитектуром *DenseNet* даје најбоље перформансе, док модел са архитектуром *LeNet5* поново даје убедљиво најлошије перформансе по свакој од три мере успешности. Архитектуре конволуционе неуронске мреже *DenseNet* и *VGG16* поново су се показале најбоље по питању перформанси. Максимална разлика у њиховим међусобним перформансама за било коју меру успешности је 2,22%.

Архитектуре конволуционих неуронских мрежа које су дале најлошије перформансе су *LeNet5* и *AlexNet*, при чему треба напоменути да је модел са архитектуром *LeNet5* имао боље перформансе него модел са архитектуром *AlexNet*.

6.2 ВРЕМЕНА ПОТРЕБНА ЗА ОБУКУ МОДЕЛА

Гледајући време потребно за обуку скупа података *ibean*, у односу на скуп података *plantaeK*, закључује се да је много више времена било потребно да би се обучио модел са подацима из скупа података *plantaeK*. Претпоставља се да је главни разлог величина слике, где се у поглављу 4.1 и 4.2 види да је величина слика у скупу података *plantaeK* од 10 до 15 пута већа од величине слике из скупа података *ibean*.

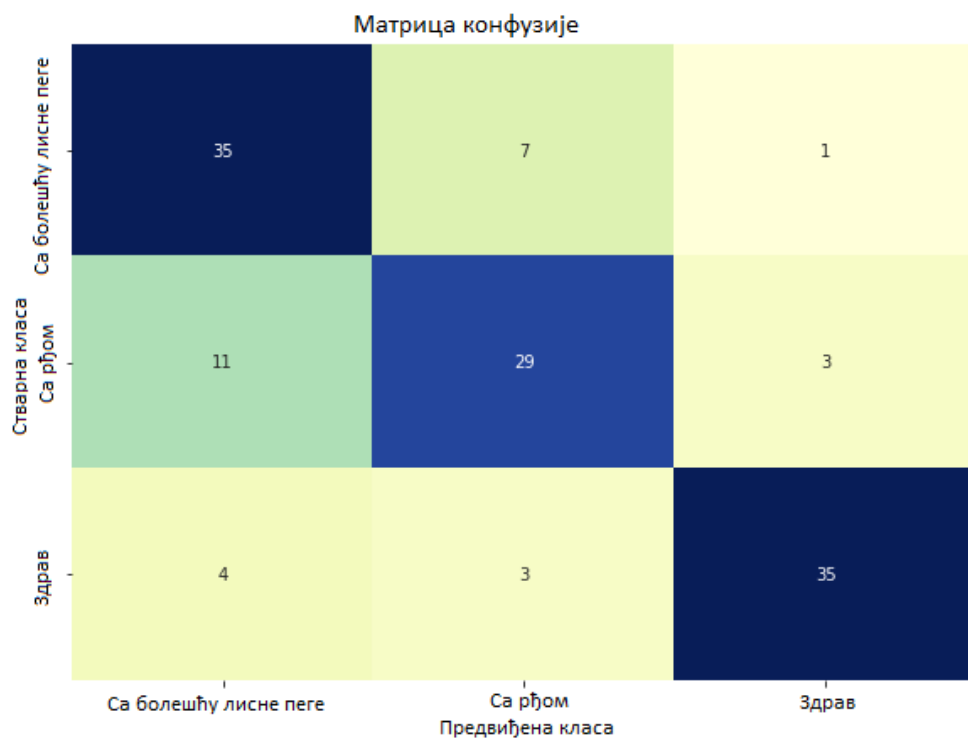
Због своје једноставности моделу са архитектуром конволуционе неуронске мреже *LeNet5* је било потребно најмање времена за обуку. Модели архитектуре *VGG16* и *DenseNet* захтевали су много времена за обуку, сходно њиховој дубини. При подешавању параметара поменутих модела потребно је било користити различите комбинације вредности параметара и обучити више модела користећи валидацију, што је допринело повећању укупног времена потребног за обуку.

6.3 ОДАБИР НАЈБОЉЕГ МОДЕЛА

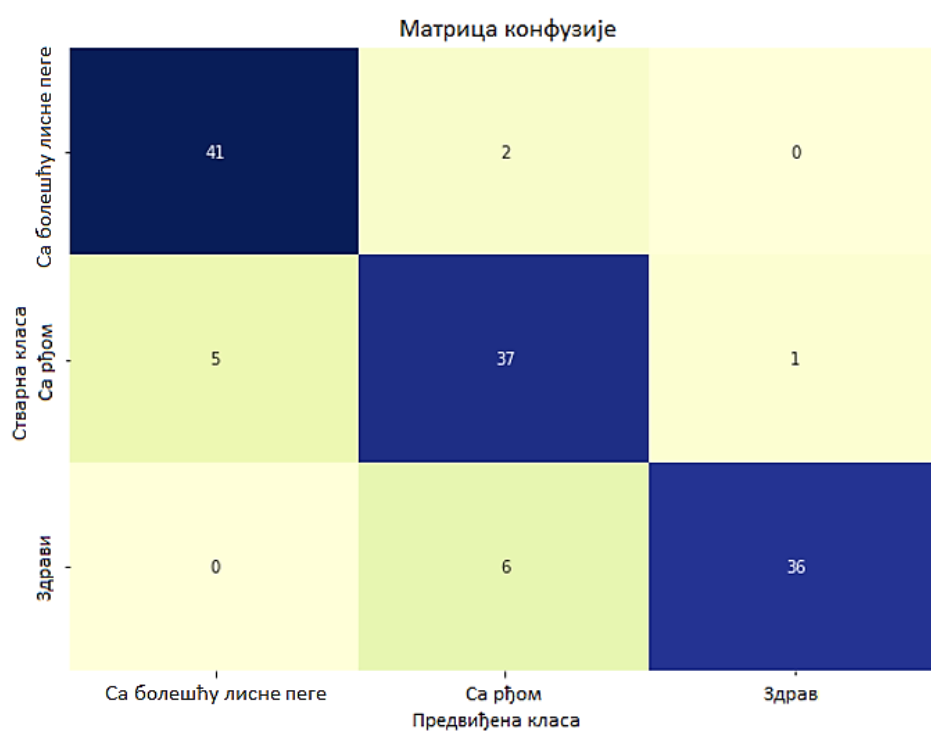
Архитектуре конволуционих неуронских мрежа *VGG16* и *DenseNet* су по посматраним параметрима дали најбоље резултате. Што се тиче првог скупа података *ibean* боље перформансе је показао модел са архитектуром конволуционе неуронске мреже *VGG16*. Са друге стране, у поглављу 6.1 је закључено је за скуп података *plantaeK* запаженије резултате у перформансама остварио *DenseNet*.

Резултати добијени за први скуп података, приказани су на *Слици 6.3.1*. Ова слика показује доминантну грешку модела архитектуре *LeNet5*, у случају када би требало се детектују листови са рђом, архитектура конволуционе неуронске мреже *LeNet5* велики број таквих листова класификује као листове са болешћу лисне пеге. Такође, осим ове грешке, мало мање уочава се и знатно слабија класификација листова, где се листови са болешћу лисне пеге класификују као листови са рђом.

За идентичне податке из тест скупа за скуп података *ibean* на *Слици 6.3.2* приказано је да је *VGG16* остварио приметно боље резултате, минимизовањем грешке предикције у оба од случајева која су горе наведена. Посебно прецизни резултати постигнути су у случају детекције здравих листова.

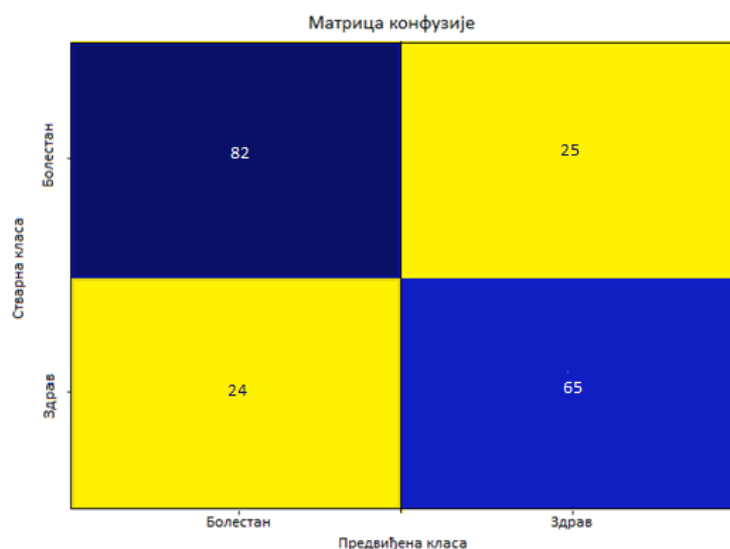


Слика 6.3.1 – Матрица конфузије модела са архитектуром конволуционе неуронске мреже LeNet5 над скупом података *ibean*



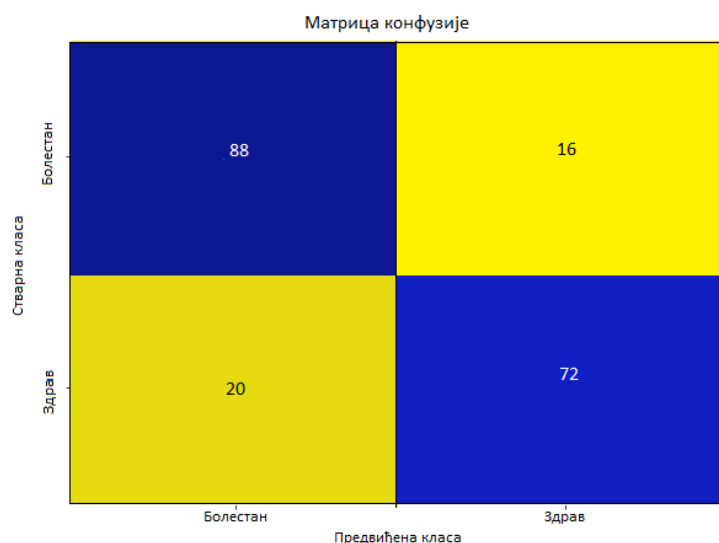
Слика 6.3.2 – Матрица конфузије модела са архитектуром конволуционе неуронске мреже VGG16 над скупом података *ibean*

За оба скупа података *ibean* и *plantaeK*, архитектура конволуционе неуронске мреже **LeNet5** дао је најлошије перформансе у односу на све остале архитектуре конволуционих неуронских мрежа. На *Слици 6.3.3* могуће је уочити добијене резултате. Изразите непрецизности у детекцији примећују се у оба случаја, здраве листове предвиђа као болесне и болесне листове предвиђа као здраве. Запажена лоша класификација здравих листова у односу на болесне.



Слика 6.3.3 – Матрица конфузије модела са архитектуром конволуционе неуронске мреже **LeNet5** над скупом података *plantaeK*

Као што је раније напоменуто, за скуп података *plantaeK* највећи проценат тачно класификованих узорака дао је модел са архитектуром конволуционих неуронских мрежа **DenseNet**. Иако се модел са архитектуром конволуционих неуронских мрежа **DenseNet** показао као најбољи у односу на остале, ипак је и код њега могуће забележити извесне грешке приликом класификације. Са *Слике 6.3.4* види се да су те грешке најчешће у случају када се здрави листови класификују у болесне.



Слика 6.3.4 – Матрица конфузије модела са архитектуром конволуционе неуронске мреже **DenseNet** над скупом података *plantaeK*

Описивани примери показују да процентуално најмање грешака праве новије архитектуре конволуционих неуронских мрежа, попут *DenseNet* и *VGG16*. Примера ради, архитектуре конволуционих неуронских мрежа *VGG16* се, уместо великог броја параметара који је типичан за старије архитектуре, фокусирају на фиксирани вредности параметара величине кернела, корака којим се врши филтрирање и вредности проширења који се примењују над сликом.

7. ЗАКЉУЧАК

Све примењене архитектуре конволуционе неуронске мреже показују лошије перформансе модела применом различитих архитектура над скупом података *plantaeK* у односу на моделе са архитектурама над скупом података *ibean*, по свим испитиваним мерама успешности. Архитектура конволуционих неуронских мрежа *VGG16* даје најбоље резултате на првом скупу података, док ни за други скуп података резултати не заостају.

Анализом перформанси модела са архитектуром конволуционих неуронских мрежа *DenseNet* може се закључити да његова природа која настоји повећању дубине конволуционе неуронске мреже, обезбеђује прихватљиве резултате, нарочито за скуп података *plantaeK*. Архитектуре које су имале значаја у историјском развоју конволуционих неуронских мрежа, у поређању са новијим архитектурама, остварују приметно лошије резултате.

Додатно, интересно истраживање које може проистећи из овог рада може бити утицај броја класа у скуповима података на перформансе сваког модела, где је могуће утврдити да ли одређени модели различитих архитектура конволуционих неуронских мрежа боље функционишу са више или мање класа. У даљем раду, могла би се користити нека друга архитектура конволуционе неуронске мреже која ће са другим вредностима параметара дати боље резултате од већ коришћених вредности. С обзиром на то да је скуп података небалансиран код скупа података *plantaeK*, могуће је покушати балансирање скупа података и видети да ли ће да се постигну боље перформансе.

На крају, могуће је користити неке друге поступке предобrade података у комбинацији с моделима класификације добијеним у овом раду, па касније поредити добијене перформансе. Могуће је у даљем раду користити претрениране моделе са познатим архитектурама конволуционих неуронских мрежа за које обука није неопходна, па би се тиме уштедело на времену за обуку.

8. БИОГРАФИЈА

Никола Савић је рођен 14. 3. 1999. у Краљеву, где је стекао своје основно и средње образовање. Школске 2018/19 године уписује се на Факултет техничких наука на студијски програм Информациони инжењеринг. Положио је све испите предвиђене планом и програмом и стекао услов за одбрану завршног рада.

9. ЛИТЕРАТУРА

- [1] ibean: Data repo for the ibean project of the AIR lab. [Online]. Available: AI-Lab-Makerere/ibean: Data repo for the ibean project of the AIR lab. (github.com) Available: <https://github.com/AI-Lab-Makerere/ibean/>.
- [2] V. P. Kour, “PlantaeK: A leaf database of native plants of Jammu and Kashmir.” Mendeley, 29-Oct-2019.
Available: <https://data.mendeley.com/datasets/t6j2h22jpx/2/>.
- [3] N. Kashive, L. Powale, and K. Kashive, “Understanding user perception toward artificial intelligence (AI) enabled e-learning,” *Int. J. Inf. Learn. Technol.*, vol. 38, no. 1, pp. 1–19, 2020.
- [4] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” *Ijeast.com*. [Online]. Available: <https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf>. [Accessed: 11-Oct-2022].
- [5] T. Nosek, B. Brkljač, D. Despotović, M. Sečujski, and T. Lončar-Turukalo, *Praktikum iz mašinskog učenja*. 2020. Fakultet tehničkih nauka, Novi Sad
- [6] D. Yi, J. Ahn, and S. Ji, “An effective optimization method for machine learning based on ADAM,” *Appl. Sci. (Basel)*, vol. 10, no. 3, p. 1073, 2020.
- [7] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” 2015.
- [8] “What is python? Executive summary,” *Python.org*. [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed: 09-Oct-2022].
- [9] “PyTorch,” *Pytorch.org*. [Online]. Available: <https://pytorch.org/>. [Accessed: 09-Oct-2022].
- [10] “pandas,” *Pydata.org*. [Online]. Available: <https://pandas.pydata.org/>. [Accessed: 29-Sep-2022].
- [11] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, “Scikit-learn: Machine learning without learning the machinery,” *GetMob. Mob. Comput. Commun.*, vol. 19, no. 1, pp. 29–33, 2015. -- SCIKLEARN
- [12] “3.3. Metrics and scoring: quantifying the quality of predictions,” *scikit-learn*. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Accessed: 11-Oct-2022].
- [13] S. Loussaief and A. Abdelkrim, “Machine learning framework for image classification,” in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, 2016.
- [14] D. Sornette, A. B. Davis, K. Ide, K. R. Vixie, V. Pisarenko, and J. R. Kamm, “Algorithm for model validation: theory and applications,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 104, no. 16, pp. 6562–6567, 2007.

- [15] H. M and S. M.n, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Min. Knowl. Manag. Process*, vol. 5, no. 2, pp. 01–11, 2015.
- [16] "Sklearn.Metrics.Balanced_accuracy_score," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html. [Accessed: 09-Oct-2022].
- [17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE Inst. Electr. Electron. Eng.*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [18] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [19] "BatchNorm2d — PyTorch 1.12 documentation," Pytorch.org. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html>. [Accessed: 11-Oct-2022].
- [20] "Max-pooling / pooling," Computersciencewiki.org. [Online]. Available: https://computersciencewiki.org/index.php/Max-pooling/_Pooling. [Accessed: 11-Oct-2022].
- [21] M. Koskela and J. Laaksonen, "Convolutional network features for scene recognition", *Processing of the ACM Conference on Multimedia*, pp. 1169-1172, 2014.
- [22] A. Elhassouny and F. Smarandache, "Trends in deep convolutional neural Networks architectures: a review," in *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, 2019.
- [23] A. G. Howard, "Some improvements on deep convolutional neural network based image classification", *CoRR*, 2013.
- [24] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-VGG16 CNN model for big data places image recognition," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", *CVPR*, 2009.
- [26] "Compose — torchvision 0.13 documentation," Pytorch.org. [Online]. Available: <https://pytorch.org/vision/stable/generated/torchvision.transforms.Compose.html>. [Accessed: 09-Oct-2022].
- [27] "Resize — Torchvision main documentation," Pytorch.org. [Online]. Available: <https://pytorch.org/vision/main/generated/torchvision.transforms.Resize.html>. [Accessed: 09-Oct-2022].
- [28] "ToTensor — Torchvision main documentation," Pytorch.org. [Online]. Available: <https://pytorch.org/vision/main/generated/torchvision.transforms.ToTensor.html>. [Accessed: 09-Oct-2022].

- [29] “Normalize — Torchvision main documentation,” Pytorch.org. [Online]. Available: <https://pytorch.org/vision/main/generated/torchvision.transforms.Normalize.html>. [Accessed: 09-Oct-2022].
- [30] “torch.utils.data — PyTorch 1.12 documentation,” Pytorch.org. [Online]. Available: <https://pytorch.org/docs/stable/data.html>. [Accessed: 09-Oct-2022]
- [31] “ImageFolder — torchvision 0.13 documentation,” Pytorch.org. [Online]. Available: <https://pytorch.org/vision/stable/generated/torchvision.datasets.ImageFolder.html>. [Accessed: 09-Oct-2022].
- [32] “Keras: the Python deep learning API,” Keras.io. [Online]. Available: <https://keras.io/>. [Pristupano: 28-Sep-2022]. -- batch_size
- [33] World cancer report: cancer research for cancer prevention, IARC Press, 2020.
- [34] “Conv2d — PyTorch 1.12 documentation,” *Pytorch.org*. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>. [Accessed: 17-Oct-2022].
- [35] “ReLU — PyTorch 1.12 documentation,” *Pytorch.org*. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html>. [Accessed: 17-Oct-2022].
- [36] “MaxPool2d — PyTorch 1.12 documentation,” *Pytorch.org*. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.MaxPool2d.html>. [Accessed: 17-Oct-2022].