

Класификација на Indoor и Outdoor фотографии

Изработиле: Ана Марија Атанасовска 206004; Алек Јармов 206012



Вовед	2
Опис на податочното множество	2
Опис на моделите.....	3
ResNet	3
Модел 1	5
Модел 2.....	5
Xception	6
Модел 3	8
Модел 4.....	9
Модел 5.....	10
Начин на тренирање.....	10
Демо Веб Апликација за предвидување на моделите	11
Резултати.....	11
ResNet	13
Модел 1	13
Модел 2.....	15
Xception	18
Модел 3	18
Модел 4.....	19
Модел 5.....	19
Заклучок.....	20
Користена литература.....	21

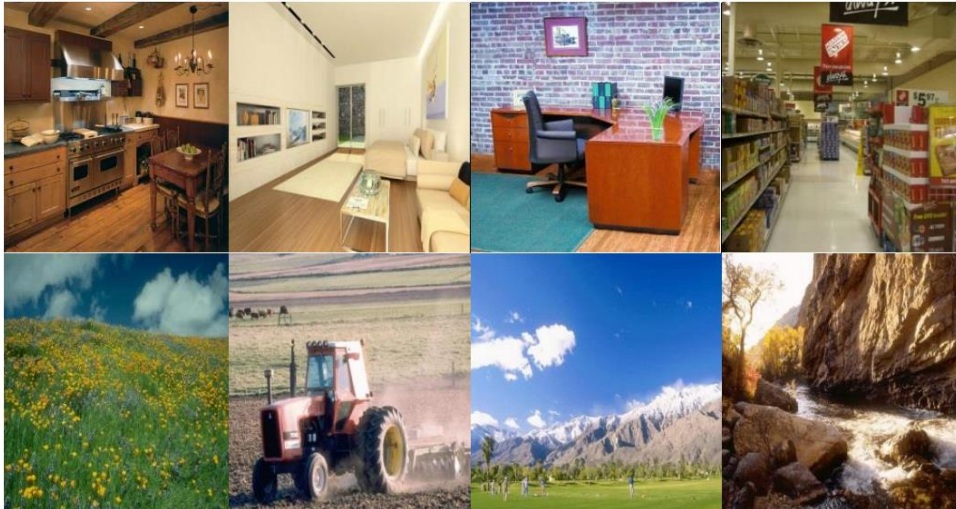
Вовед

Способноста автоматски да се класифицираат слики во внатрешна и надворешна (indoor и outdoor) категорија е од големо значење за различни апликации. Им овозможува на интелигентните системи да го разберат контекстот на сликите и да го адаптираат нивното однесување во согласност со тоа. На пример, во роботика различните начини за навигирање можат да се одлучат врз база на тоа во каква околина се движи роботот, дали во надворешна или внатрешна. Слично, при пребарување на слики корисникот може да преферира да пребарува во внатрешен или надворешен простор. Можеме да заклучиме дека потребни се точни и ефикасни алгоритми за класификација за да се овозможат вакви апликации.

Во овој проект ќе ги истражime и споредиме различните алгоритми за класификација на фотографии во ‘indoor’ и ‘outdoor’ класа. Ќе земеме во предвид различни техники, во главно transfer learning со одредени модели за длабоко учење. Со евалуација на перформансот на овие алгоритми на избрано податочно множество, ќе целиме кон тоа да го идентификуваме методот кој е најсоодветен за нашиот проблем и достигнува висока класификациска точност и робустност. Би користеле различни метрики за евалуација, како точност, прецизност, recall и F1 score. Ќе употребиме и дадени примероци и ќе увидиме како се класифицираат од страна на моделите.

Опис на податочното множество

Го искористивме податочното множество достапно на [линкот](#). [1] Ова податочно множество е лабелирано, поделено е во 2 групи на слики, indoor и outdoor слики, кое што одговара на проблемот кој го обработуваме во овој проект. Податочното множество се состои од вкупно 800 слики, 400 слики кои се однесуваат на сцени од внатрешен простор и 400 слики кои се однесуваат на сцени од надворешен простор што значи дека е балансирано.



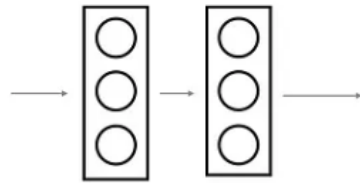
На оваа слика се прикажани некои од сликите во податочното множество, сликите од горниот ред се од класата 'indoor', додека пак сликите од долниот ред се од класата 'outdoor'.

Опис на моделите

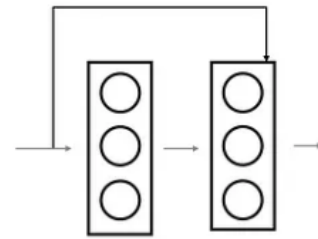
ResNet

ResNet е кратенка за Residual Networks и е класична невронска мрежа користена за многу задачи од компјутерска визија. Овој модел беше победник на ImageNet предизвикот во 2015. ResNet ни дозволува да тренираме екстремно длабоки невронски мрежи со 150+ слоеви успешно. Но, зголемувањето на длабочината на мрежата не работи со само натрупување на слоеви заедно. Длабоките мрежи се тешки за тренирање поради проблемот на vanishing gradient, со пропација на gradientот назад до пораните слоеви повторуваната мултипликација може да го направи gradient-от екстремно мал и како резултат како што мрежата оди се подлабоко, нејзиниот перформанс се заситува или почнува да се намалува рапидно. Силна страна на ResNet е концептот на прескокнување на конекција (skip connection). Овој концепт вклучува натрупување на конволуциски слоеви како претходно, но сега исто го додаваме оригиналниот инпут на излезот од конволуцискиот блок. Skip Connections работи бидејќи го ублажува проблемот на vanishing gradient со дозволување на овој алтернативен пат на gradient-от да протече и бидејќи дозволува моделот да научи функција на идентитет што се осигура дека повисокиот слој ќе биде барем исто добар како понискиот слој, не полош. На Слика 1 е прикажано како изгледа Skip Connection концептот. [2]

without skip connection

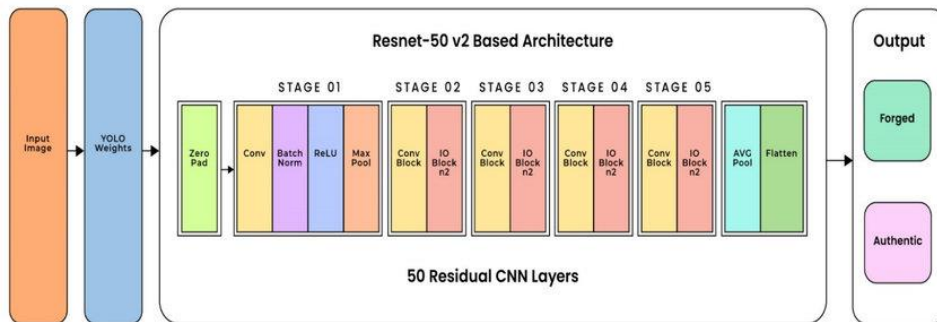


with skip connection



Слика 1. Skip Connection во ResNet

За целите на нашето истражување ќе направиме наш модел од длабоко учење користејќи го ResNet50V2 како слој. ResNet50V2 прима слики со големина 224x224 пиксели во 3 канали. Што значи ќе треба да го направиме соодветното предпроцесирање на сликата за да биде бараните димензии. На следната слика е прикажана архитектурата на ResNet50V2.



Модел 1

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
resnet50v2 (Functional)	(None, 7, 7, 2048)	23564800
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12845184
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 64)	4160
dense_3 (Dense)	(None, 2)	130

```
=====  
Total params: 36,422,530  
Trainable params: 36,377,090  
Non-trainable params: 45,440  
=====
```

Ќе искористиме Sequential модел со следните слоеви:

- ResNet50V2 слој
- Flatten слој - слој кој го порамнува влезот, на пример доколку влезот е во форма (None, 1, 10, 64) ќе го трансформира во (None, 640) [3]
- Dense слоеви - слој од неврони во кој секој неврон добива влез од сите неврони од претходниот слој. [4] Ние во нашата архитектура ќе користиме 4 Dense слоеви од по 128, 64, 64 и 2 неврон.

Овој модел ќе биде трениран во 100 епохи.

Модел 2

Направивме уште еден модел кој што се базира на ResNet50V2, но овој пат користиме и тн. Dropout слоеви. Целосната архитектура на овој модел е прикажана на следната слика.

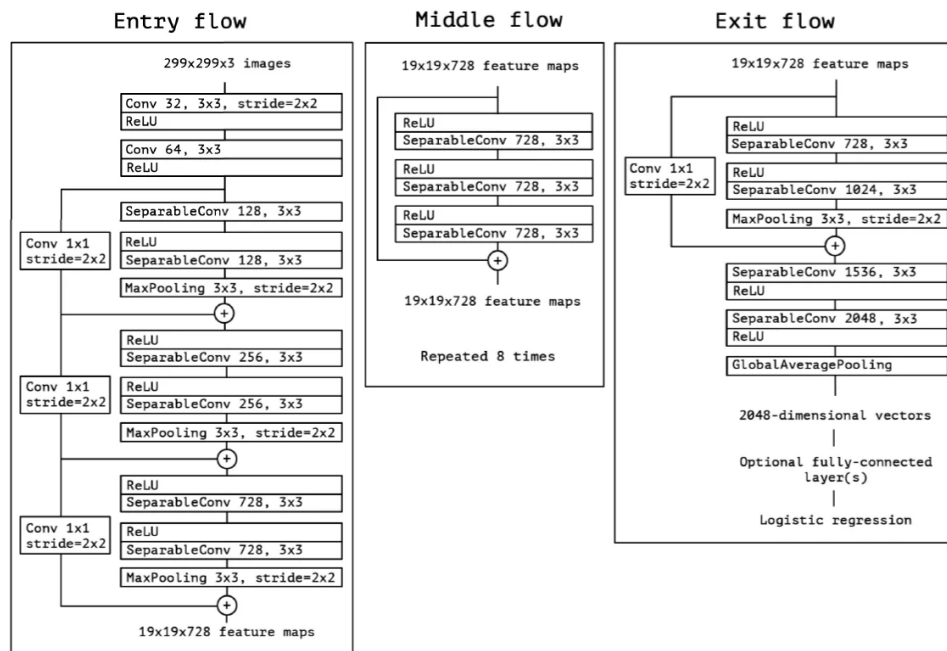
Layer (type)	Output Shape	Param #
resnet50v2 (Functional)	(None, 7, 7, 2048)	23564800
flatten_2 (Flatten)	(None, 100352)	0
dense_4 (Dense)	(None, 264)	26493192
dropout (Dropout)	(None, 264)	0
dense_5 (Dense)	(None, 128)	33920
dropout_1 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 1)	129
Total params: 50092041 (191.09 MB)		
Trainable params: 50046601 (190.91 MB)		
Non-trainable params: 45440 (177.50 KB)		

Овој модел е различен од претходниот во формата на Dense слоевите, претходно имавме 4 последователни Dense слоеви со 128,64,64 и 2 неврони, додека пак сега имаме 3 слоеви со 264,128 и 1 неврони, кои се одвоени со Dropout слоеви со 264 и 128 неврони соодветно и со dropout rates 0.3 и 0.1 соодветно. Dropout слојот е слој што случајно поставува дел од влезните единици со големина на dropout rate на 0 на секој чекор за време на тренирањето на моделот што помага да се превенира overfitting. Влезовите кои нема да се постават на 0 се скалираат со $1/(1-\text{ратата})$. [5]

Слично како и претходниот, и овој модел ќе го тренираме во 100 епохи.

Xception

Xception е модел предложен од Francois Chollet. Претставува продолжение на Inception архитектурата. [9] Inception архитектурата е воведена од страна на истражувачи од Google во 2014 и е најголемата и најефикасна архитектура за длабоко учење од тоа време. Се состои од компоненти кои се повторуваат наречени Inception модули. [10] Архитектурата на Xception модел е прикажана на Слика 2.



Слика 2. Архитектура на Xception

Моделот Xception над валидациското податочно множество на ImageNet има Top-1 точност од 79%, а Top-5 од 94.5% [11] Архитектурата на Xception може да се види и со извршување на следните наредби:

```
from keras.applications.xception import Xception

# Create an instance of the Xception model
model = Xception(weights='imagenet')

# Display the model summary
model.summary()
```

Излезот би го имал следниот изглед:

```

Model: "xception"
-----
Layer (type)                Output Shape              Param #
=====
input_1 (InputLayer)        [(None, 299, 299, 3)]    0
-----
block1_conv1 (Conv2D)        (None, 149, 149, 32)     864
-----
block1_conv1_bn (BatchNormal (None, 149, 149, 32)    128
-----
block1_conv1_act (Activation (None, 149, 149, 32)    0
-----
...
...
...
<more layers>
...
...
...
-----
avg_pool (GlobalAveragePooli (None, 2048)             0
-----
predictions (Dense)         (None, 1000)             2049000
=====
Total params: 22,910,480
Trainable params: 22,855,952
Non-trainable params: 54,528
-----

```

Ние ќе го користиме овој модел со хиперпараметрите, односно тежините научени од податочното множество на ImageNet.

Модел 3

Хсептион ќе го користиме повторно како слој во Sequential невронска мрежа. Овој пат нема да користиме Dropout слоеви па целиот модел кој ќе го пробаме, ќе ја има следната архитектура:

Model: "sequential"

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 256)	25690368
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 2)	130

=====
Total params: 46,593,130
Trainable params: 46,538,602
Non-trainable params: 54,528

Моделот ќе го тренираме во 100 епохи.

Модел 4

Сакаме да го пробаме Xception слојот во комбинација на Dropout слоеви и да го споредиме со претходниот модел. И овој модел ќе го тренираме во 100 епохи и ќе ја има следната архитектура:

Layer (type)	Output Shape	Param #
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten_1 (Flatten)	(None, 100352)	0
dense_3 (Dense)	(None, 128)	12845184
dropout_2 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 2)	130

=====
Total params: 33,715,050
Trainable params: 33,660,522
Non-trainable params: 54,528

Како работат Dropout слоевите и нивното значење е опишано во делот каде што се опишува Модел 2.

Модел 5

Сега сакаме да пробаме дополнително предпроцесирање на сликата дали ќе доведе до подобри резултати. Освен што ги менуваме димензиите на сликата, овој модел исто така ја зголемува и нејзината сатурација како прв предпроцесирачки чекор. Идејата е топлите бои да влијаат помалце при класификацијата. Сатурацијата е ставена на 0.75 во скала од [0,1] каде што 0 значи дека сликата е целосно црно-бела, а 1 значи дека е целосно сатурирана. Исто така сега направивме ратата во Dropout слојот да соодветствува со бројот на неврони односно за слојот со 256 неврони ратата е 0.4, за слојот со 128 неврони ратата е 0.2 и конечно за слојот со 64 неврони ратата е 0.1

Layer (type)	Output Shape	Param #
random_saturation_1 (Random Saturation)	(None, 224, 224, 3)	0
xception (Functional)	(None, 7, 7, 2048)	20861480
flatten_3 (Flatten)	(None, 100352)	0
dense_9 (Dense)	(None, 256)	25690368
dropout_6 (Dropout)	(None, 256)	0
dense_10 (Dense)	(None, 128)	32896
dropout_7 (Dropout)	(None, 128)	0
dense_11 (Dense)	(None, 64)	8256
dropout_8 (Dropout)	(None, 64)	0
dense_12 (Dense)	(None, 2)	130
=====		
Total params: 46,593,130		
Trainable params: 46,538,602		
Non-trainable params: 54,528		

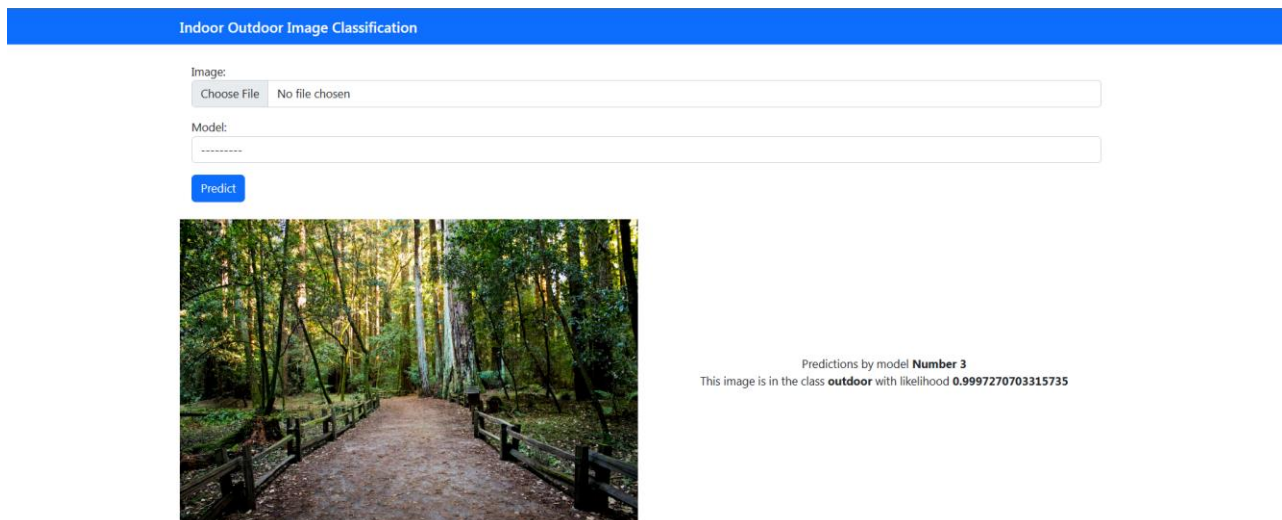
Моделот беше трениран на 200 епохи.

Начин на тренирање

Сите споменати модели ќе ги тренираме на споменатото податочно множество кое ќе го поделиме на следниот начин 80% тренирачки дел и 20% валидациски дел. Сликите ќе се читаат со користење на функцијата `tf.keras.utils.image_dataset_from_directory` која ја менува нивната големина при самото читање на сликата.

Демо Веб Апликација за предвидување на моделите

Во рамките на овој проект направивме апликација каде што корисниците може да ги тестираат нашите модели. За изработка на оваа веб апликација ја искористивме Python рамката за развој на вакви апликации Django, а за визуелниот интерфејс алатката Bootstrap. На оваа веб апликација може да се прикачи слика, да се избере еден од 5те модели опишани претходно и да се добие класата на која што припаѓа избраната слика според избраниот модел. Изгледот на оваа веб страница е прикажан на следната фотографија:



Резултати

Во овој дел ќе ги прикажеме резултатите од тестирањето на опишаните модели во претходната секција. За евалуација на моделите ќе ги искористиме следните метрики:

- **Binary Cross Entropy:** уште позната како `log loss`, е метрика што ја мери разликноста помеѓу предвидената веројатносна дистрибуција и вистинските

бинарни лабели на податочното множество. Binary cross entropy ја споредува секоја од предвидените веројатности со вистинските класни излези кои можат да бидат или 0 или 1. Потоа се калкулира score што ги пенализира веројатностите базирано на растојанието од очекуваната вредност. Ја има следната формула [6]

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^N - (y_i * \log(p_i) + (1-y_i) * \log(1-p_i))$$

Слика 2. Binary Cross Entropy

- **Точност (Accuracy):** во проблем на бинарна класификација како нашиот, ако со TP ги означиме вредностите кои се предвидени како 1 и навистина се такви, со TN ги означиме вредностите кои се предвидени како 0 и навистина се такви, со FP ги означиме вредностите кои се предвидени како 1, но се 0 и со FN оние кои се предвидени како 0, но се 1, се пресметува на следниот начин [7]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Слика 3. Точност

- **Прецизност (Precision):** доколку ги означиме вредностите на сличен начин како и во Accuracy со TP, TN, FP и FN, прецизността би се пресметувала на следниот начин: $TP/(TP+FP)$ што е во суштина од сите предвидени вредности 1, колку се навистина 1. [8]
- **Recall:** оваа вредност укажува на бројот на точно предвидени позитивни вредности од сите позитивни вредности односно има вредност $TP/(TP+FN)$. [8] Имено ако нема елементи кои припаѓаат на класата 1 предвидени како да припаѓаат на класата 0 од страна на моделот, recall-от би изнесувал 1. [8]
- **F1 Score:** ги комбинира и прецизността и recall-от во една метрика која ги искажува и двете својства и ја има формата $(2*Precision*Recall)/(Precision + Recall)$. [8]

Би сакале да забележиме дека ние во нашето податочно множество како класа 1 ќе ја сметаме класата на слики во надворешен простор, а како класа 0 слики во внатрешен простор.

По пресметувањето на споменатите метрики, моделите ќе ги испробаме на неколку тест примери надвор од податочното множество споменато во делот “Опис на податочното множество”.

ResNet

Модел 1

Првиот модел кој го изработивме со користење на ResNet слој покажа солидни резултати над валидациското множество. Целосен извештај за пресметаните метрики над валидациското множество е достапен во Табела 1. Можеме да забележиме висок Recall што укажува на тоа дека најголем дел од сликите од outdoor класата се точно предвидени.

Име на метрика	Вредност
Binary Cross Entropy	0.4332
Точност	0.8938
Прецизност	0.8938
Recall	0.8938
F1 score	0.8937

Табела 1. Вредност на метрики за моделот 1 пресметани при апликација на моделот над валидациското множество



Predictions by model **Number 1**
This image is in the class **indoor** with likelihood **0.9975470900535583**

Како што можеме да видиме оваа слика моделот ја предвидува лошо, вели дека припаѓа во класата indoor. Исто и оваа слика:



Predictions by model **Number 1**
This image is in the class **indoor** with likelihood **0.9975699782371521**

Но оваа слика ја предвидува во ред, со indoor класа.



Predictions by model **Number 1**
This image is in the class **indoor** with likelihood **0.9975470900535583**

Модел 2

Вториот модел кој го изработивме со користење на ResNet слој покажа 91.25% точност над валидациското множество. Сите метрики кои ги тестиравме над моделот се прикажани во Табела 2. Во споредба со моделот 1 забележуваме повисока точност, прецизност и recall, пониска binary cross ентропија, но понизок F1 score. Можеме да забележиме дека во споредба со моделот 1 Dropout слоевите, како и целосната архитектура на моделот се покажаа како успешни за решавање на нашиот проблем и превенција на overfitting.

Име на метрика	Вредност
Binary Cross Entropy	0.3524
Точност	0.9125
Прецизност	0.9036
Recall	0.9259
F1 score	0.6722

Табела 2: Вредност на метрики за моделот 2 пресметани при апликација на моделот над валидациското множество

Некои нови слики(кои што воопшто ги нема во податочното множество) кои што грешно ги класифицира моделот се следниве:



Predictions by model **Number 2**
This image is in the class **indoor** with likelihood **0.9997536041482817**

Иако оваа слика е од класата 'outdoor', моделот ја има класифицирано како 'indoor', но оваа слика е малце тешко да се класифицира поради мебелот и боите



Predictions by model **Number 2**
This image is in the class **indoor** with likelihood **0.9995968030125368**

Оваа иако е појасно дека е 'outdoor' најверојатно поради кафеавите бои е погрешно класифицирана

Следната слика точно ја класифицира



Predictions by model **Number 2**
This image is in the class **indoor** with likelihood **0.9936814662069082**

Xception

Модел 3

Метриките од третиот модел се дадени во табела 3.

Име на метрика	Вредност
Binary Cross Entropy	0.2453
Точност	0.95
Прецизност	0.95
Recall	0.95
F1 score	0.9499

Табела 3. Вредност на метрики за моделот 3 пресметани при апликација на моделот над валидациското множество

Овој модел го тестиравме и на примерокот фотографиии спомнати претходно. Има исто однесување, ги класифицира во истата класа сите слики освен оваа, за која дава точно предвидување:



Predictions by model **Number 3**
This image is in the class **outdoor** with likelihood **0.6333973407745361**

Модел 4

Метриките од четвртиот модел се дадени во табела 5. Може да забележиме дека овој пат Dropout слоевите доведоа до полоши резултати за разлика од моделите ко што користеа ResNet.

Име на метрика	Вредност
Binary Cross Entropy	0.2453
Точност	0.8875
Прецизност	0.8875
Recall	0.8875
F1 score	0.8864

Табела 4. Вредност на метрики за моделот 4 пресметани при апликација на моделот над валидациското множество

Овој модел предвидува исто како првите 2 модела односно единствена слика што ја предвидува точно е



Predictions by model **Number 4**
This image is in the class **indoor** with likelihood **0.9992096424102783**

Модел 5

Можеме да забележиме дека претходните модели грешат на сликата со висока сатурација. Овој модел како што напоменавме има слој кој ги сатуира сите фотографии,

се со цел да го ублажи нејзиното влијание врз предвидувањето. Метриките за овој модел се дадени во Табела 5. Може да заклучиме дека дополнителното предпроцесирање на сликите така што им додаваме сатурација не доведе до подобрување на моделот и како најдобар модел може да се издвои третиот.

Име на метрика	Вредност
Binary Cross Entropy	0.4578
Точност	0.9312
Прецизност	0.9312
Recall	0.9312
F1 score	0.9309

Табела 5. Вредност на метрики за моделот 4 пресметани при апликација на моделот над валидациското множество

Може да се забележи дека иако оваа слика ,што сите други модели исто така ја класифицираат точно, и тој ја класифицира со најмала сигурност.



Predictions by model **Number 5**
This image is in the class **indoor** with likelihood **0.513188362121582**

Заклучок

Овој проект прикажа исцрпна анализа на алгоритми за класификација на слики во indoor и outdoor класи. Целта ни беше да го идентификуваме најефективниот пристап за добивање точни и робустни класификациски резултати за оваа специфична задача.

Споредивме 5 keras модели вклучувајќи ги Xception и ResNet50V2 моделот. Секој алгоритам беше евалиуран со користење на метриките точност, прецизност, recall и F1 Score.

Резултатите на нашето истражување укажуваат дека Xception дава подобар перформанс во однос на другите алгоритми при решавање на проблемот на класификација на слики во indoor и outdoor класи. Моделот кој го користеше Xception моделот како слој се покажа како способен ефективно да разликува внатрешни и надворешни сцени, овозможувајќи вредни информации за контекстот на сликите.

Користењето на претренирани модели како споменатите е пример за Transfer Learning техника и ни дозволува да го искористиме максимално знаењето добиено од обемни податочни множества, подобрувајќи го перформансот на моделот во indoor-outdoor задачата на класификација.

Важно е да напоменеме дека перформансот на алгоритмите може да варира во зависност на карактеристиките на податочното множество, големината на тренирачкото множество, но и комплексноста на сцените кои се класифицираат. Така, од огромно значење беше да се избере најсоодветниот алгоритам базирано на специфичните барања и ограничувањата на апликацијата.

Резултатите од нашето истражување допринесуваат на полето за класификација на слики со прикажување на силните и слабите страни на искористените алгоритми. Во понатамошната работа би можеле да ги комбинираме споменатите алгоритми, но и да се аплицираат додатни претпроцесирачки алгоритми над нашето податочно множество.

Користена литература

1. Nadian-Ghomsheh, Ali (2017). Indoor-Outdoor dataset. figshare. Dataset.
<https://doi.org/10.6084/m9.figshare.4595323.v1>
2. [Understanding and Coding a ResNet in Keras | by Priya Dwivedi | Towards Data Science](#)
3. [Flatten layer \(keras.io\)](#)
4. [Introduction to Convolutional Neural Network \(CNN\) using Tensorflow | by Govinda Dumane | Towards Data Science](#)
5. [Dropout layer \(keras.io\)](#)
6. [Binary Cross Entropy/Log Loss for Binary Classification \(analyticsvidhya.com\)](#)
7. [Accuracy and precision - Wikipedia](#)
8. [How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification - MachineLearningMastery.com](#)
9. [Image Recognition using Pre-trained Xception Model in 5 steps | by Gopalakrishna Adusumilli | Analytics Vidhya | Medium](#)
10. [Deep Learning: Understanding The Inception Module | by Richmond Alake | Towards Data Science](#)

11. [Keras Applications](#)