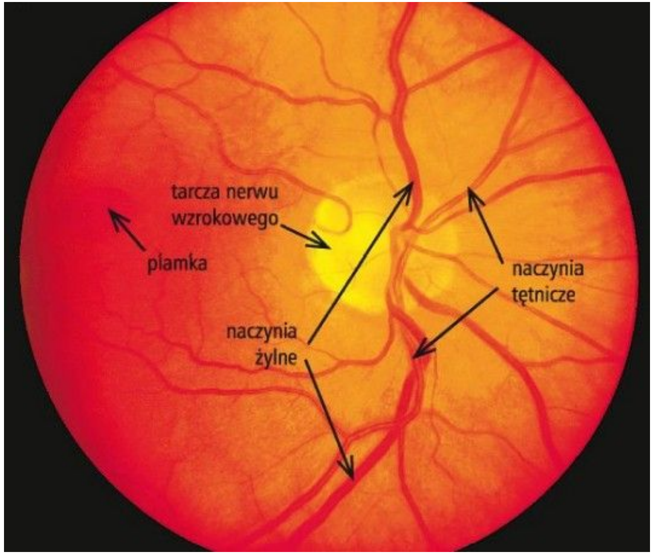
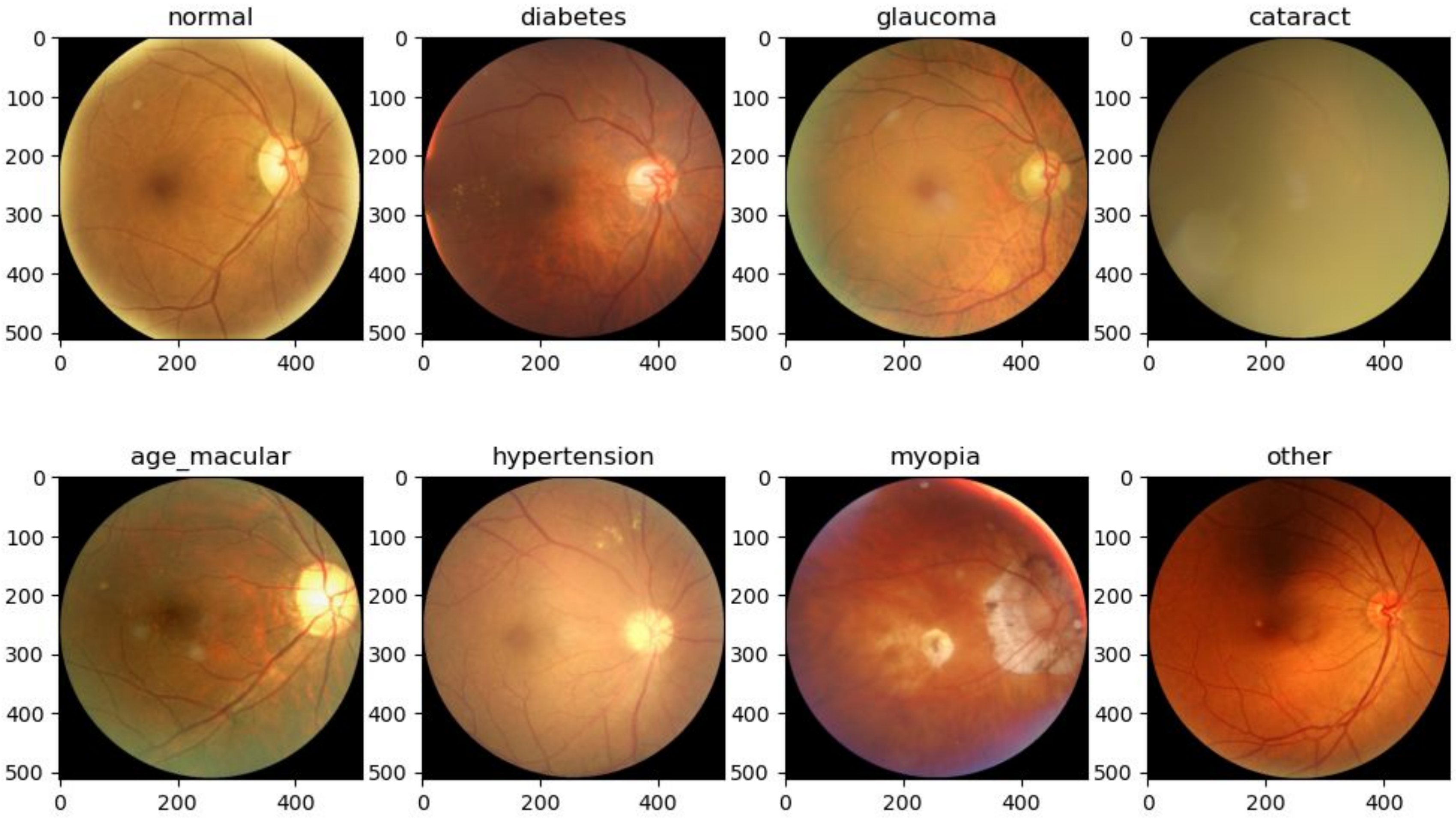
Ocular disease recognition

Jakub Szpunar

Temat

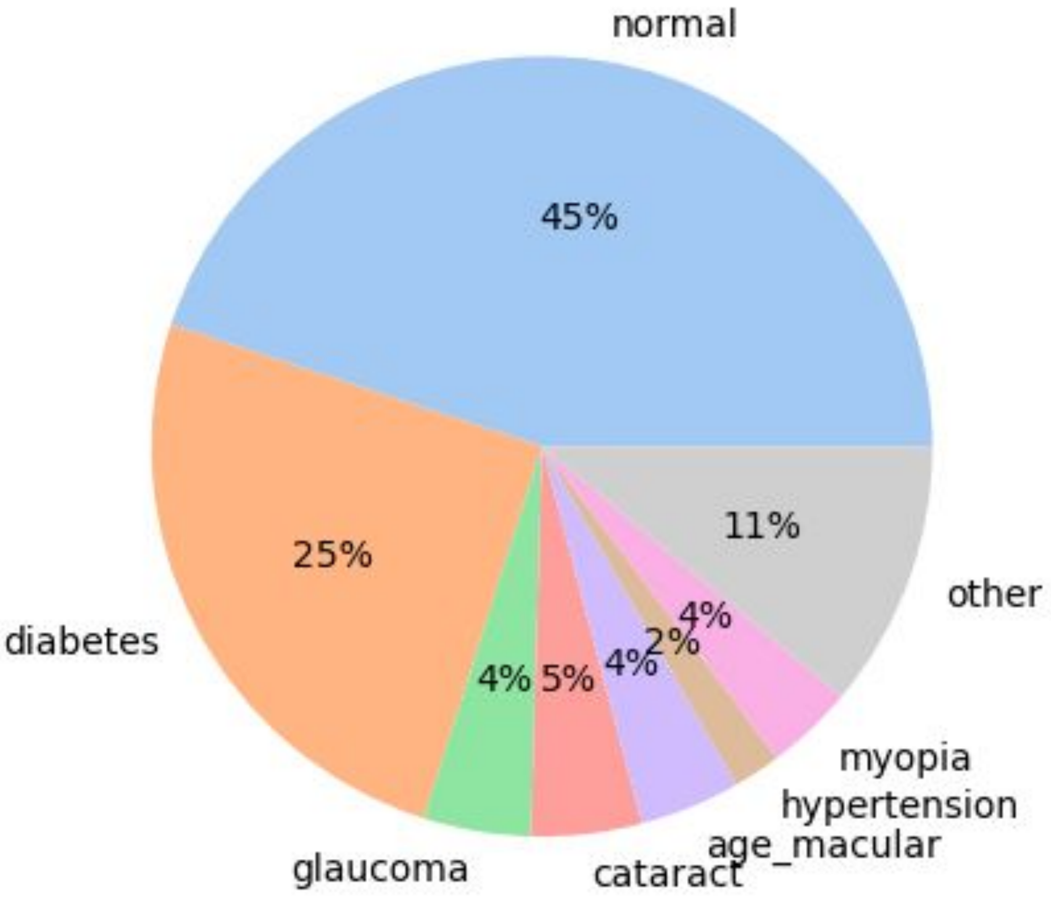
Klasyfikacja wieloklasowa chorób oczu na podstawie zdjęć dna oka przy pomocy konwolucyjnej sieci neuronowej.





normal
diabetes
glaucoma
cataract
age_macular
hypertension
myopia
other

Ocular Disease data distribution



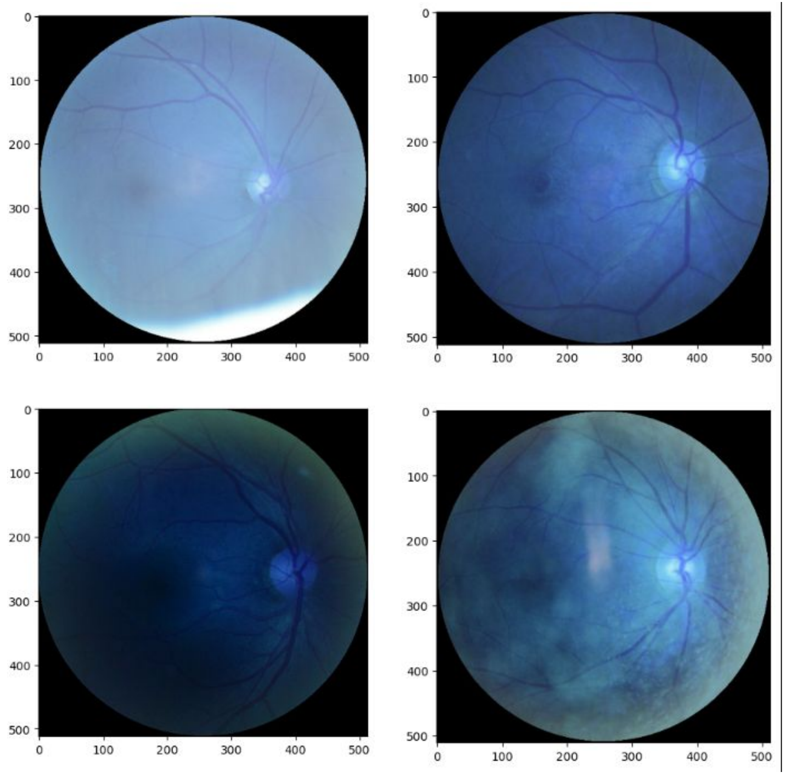
Preprocessing danych

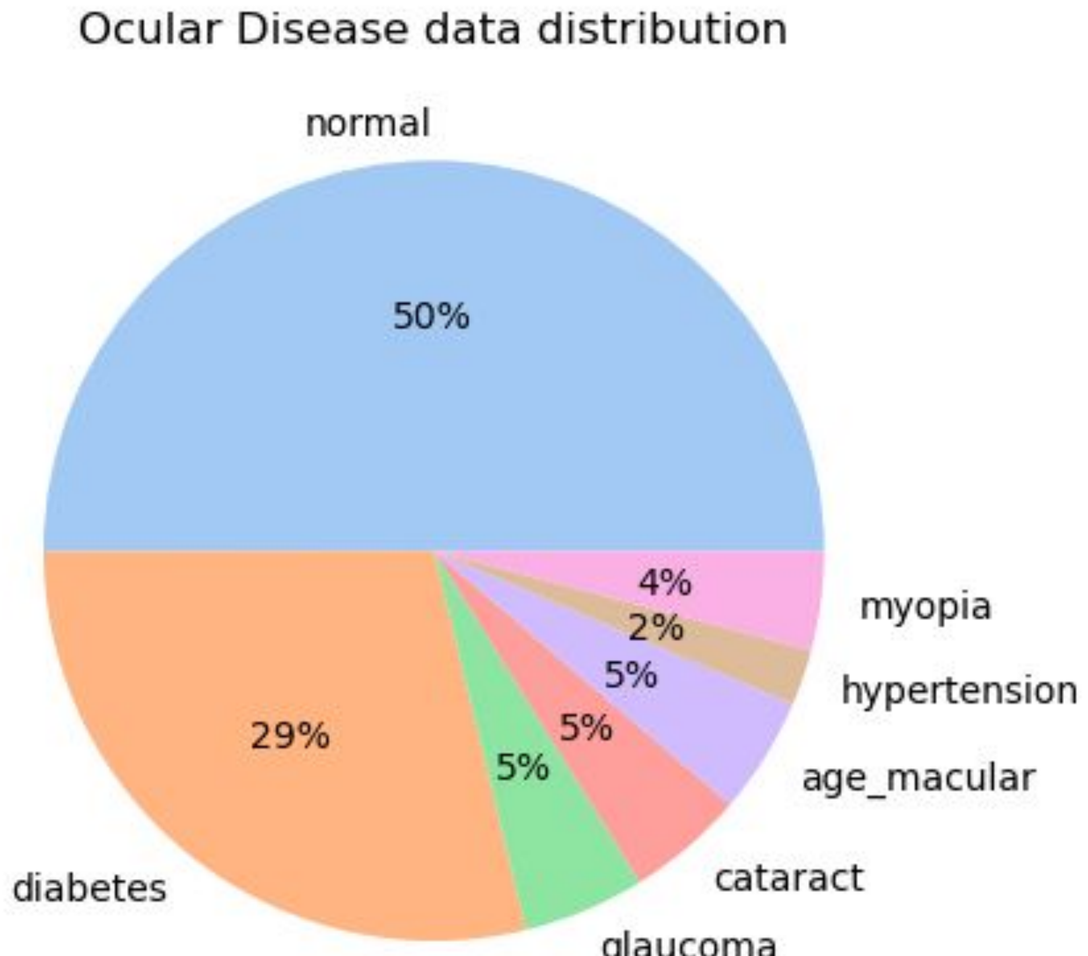
```
'hypertensive retinopathy, age-related macular degeneration',
'epiretinal membrane, moderate non proliferative retinopathy, laser spot',
'vitreous degeneration, lens dust', 'low image quality,maculopathy',
'moderate non proliferative retinopathy, lens dust, drusen',
'mild nonproliferative retinopathy, lens dust, drusen',
'wet age-related macular degeneration, myopia retinopathy',
'macular epiretinal membrane, laser spot',
'epiretinal membrane, epiretinal membrane, lens dust',
```

```
'mild nonproliferative retinopathy, vitreous degeneration',
'vitreous degeneration, mild nonproliferative retinopathy',
'optic disk photographically invisible',
'moderate non proliferative retinopathy, chorioretinal atrophy',
'epiretinal membrane over the macula, white vessel',
'laser spot, white vessel',
```

Pozbycie się problematycznych zdjęć oraz klasy ‘other’
+
resize z 512x512 do 256x256

Lens dust




Ocular Disease data distribution
normal
50%
29%
diabetes
5%
glaucoma
5%
cataract
5%
age_macular
2%
hypertension
4%
myopia

Pierwszy prosty model

```
[21]: model1 = models.Sequential()
      model1.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 3)))
      model1.add(layers.MaxPooling2D((2, 2)))
      model1.add(layers.Conv2D(64, (3, 3), activation='relu'))
      model1.add(layers.MaxPooling2D((2, 2)))
      model1.add(layers.Conv2D(128, (3, 3), activation='relu'))
      model1.add(layers.MaxPooling2D((2, 2)))
      model1.add(layers.Flatten())
      model1.add(layers.Dense(64, activation='relu'))
      model1.add(layers.Dense(num_class, activation='softmax'))
```

Accuracy jako metryka przy niezbalansowanym datasecie nie jest dobrym wyborem. Lepszym wyborem jest Recall, Precision lub F1

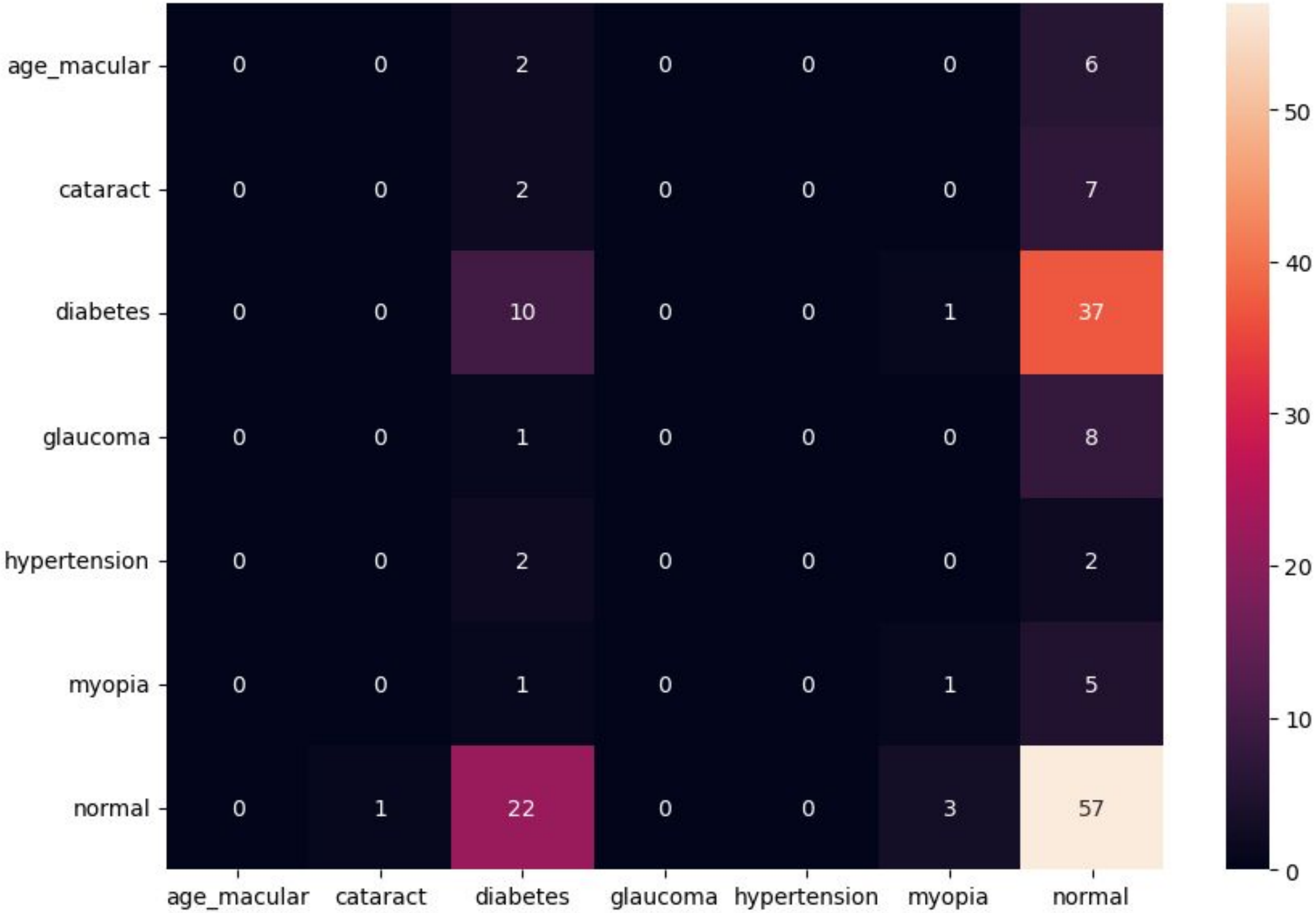
```
[25]: result1 = model1.evaluate(test_generator)
      dict(zip(model1.metrics_names, result1))

      193/193 [==============================] - 8s 40ms/step - loss: 1.4821 - acc: 0.4896

[25]: {'loss': 1.4821242094039917, 'acc': 0.48960497975349426}
```

Accuracy prawie ~48% przy 8 klasach nie jest najgorszym wynikiem dla tak prostego modelu biorąc pod uwagę użyty dataset, lecz trzeba zauważyć, że aż 50% stanowią obrazy klasy ‘normal’.

Użycie bardziej rozbudowanego modelu

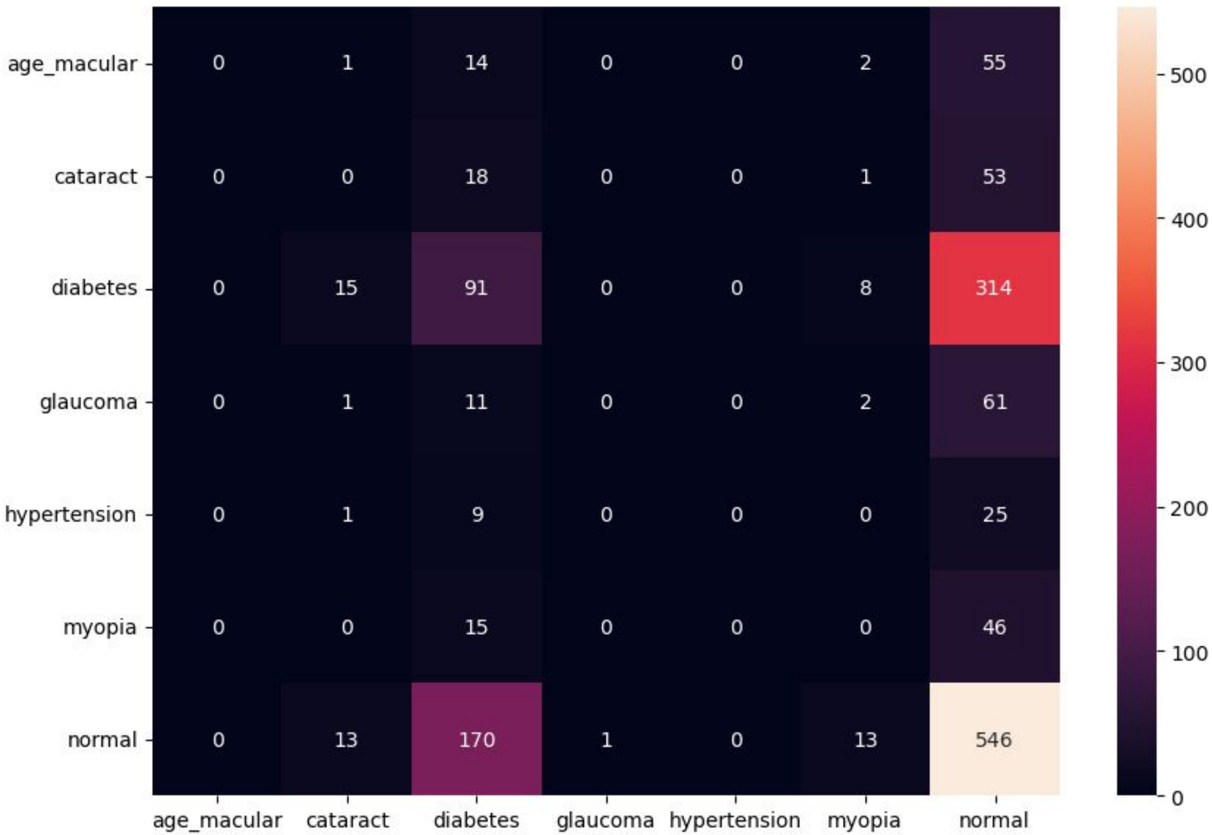
```
[34]: model8 = models.Sequential()
model8.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 3)))
model8.add(layers.MaxPooling2D((2, 2), padding='same'))
model8.add(layers.Conv2D(128, (3, 3), activation='relu'))
model8.add(layers.Conv2D(256, (3, 3), activation='relu'))
model8.add(layers.MaxPooling2D((2, 2), padding='same'))
model8.add(layers.Conv2D(512, (3, 3), activation='relu'))
model8.add(layers.Conv2D(256, (3, 3), activation='relu'))
model8.add(layers.MaxPooling2D((2, 2), padding='same'))
model8.add(layers.Flatten())
model8.add(layers.Dropout(0.5))
model8.add(layers.Dense(256, activation='relu'))
model8.add(layers.Dense(7, activation='softmax'))

model8.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
              optimizer=tf.keras.optimizers.Adam(),
              metrics=[tf.keras.metrics.Recall(), tf.keras.metrics.Precision()])
```

Rezultat

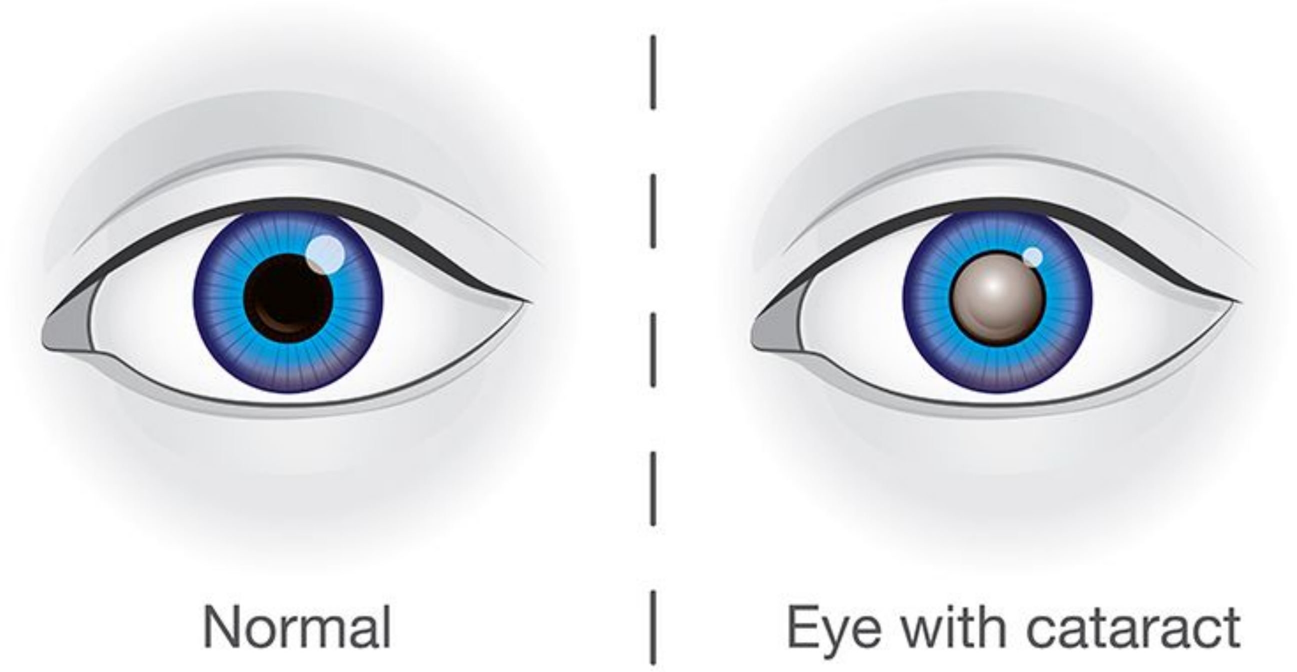


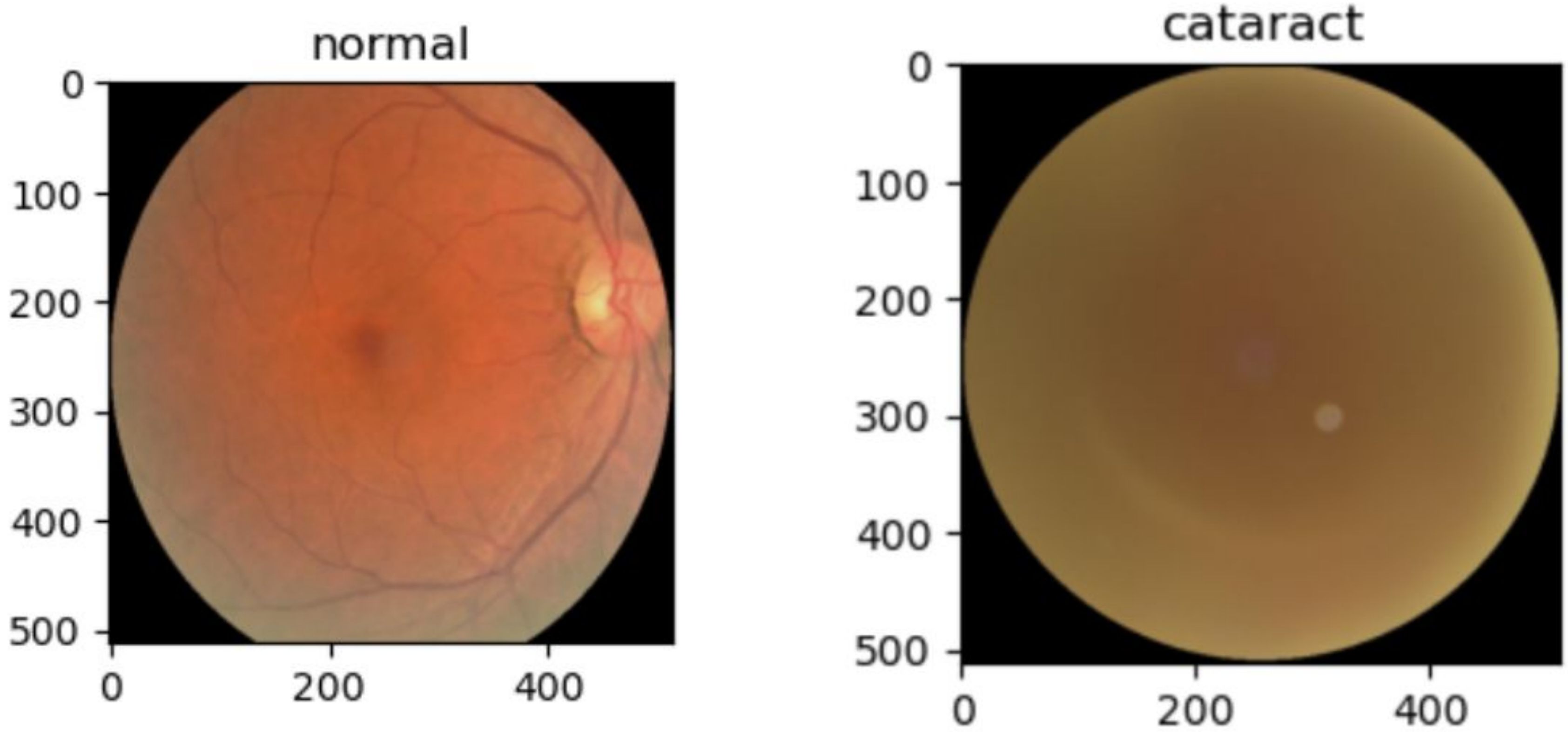
age_macular
cataract
diabetes
glaucoma
hypertension
myopia
normal
0 0 2 0 0 0 6
0 0 2 0 0 0 7
0 0 10 0 0 1 37
0 0 1 0 0 0 8
0 0 2 0 0 0 2
0 0 1 0 0 1 5
0 1 22 0 0 3 57
age_macular
cataract
diabetes
glaucoma
hypertension
myopia
normal
50
40
30
20
10
0

Test na zbiorze walidacyjnym



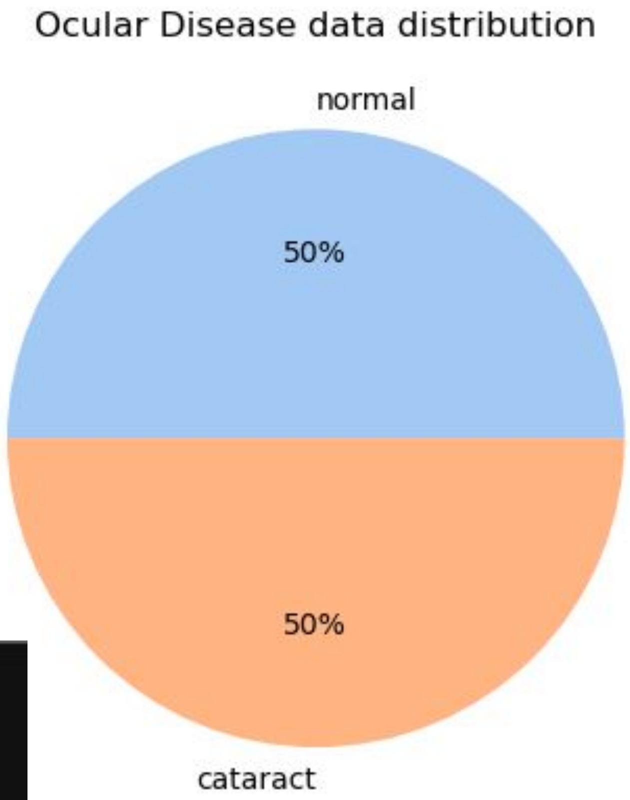
Ograniczenie liczby klas - sprawdzenie jak modele radzą sobie z klasyfikacją binarną




normal
cataract
0
100
200
300
400
500
0
200
400

Ograniczenie zbioru - równa liczebność klas

```
['normal', 'cataract']
[269, 269]
```

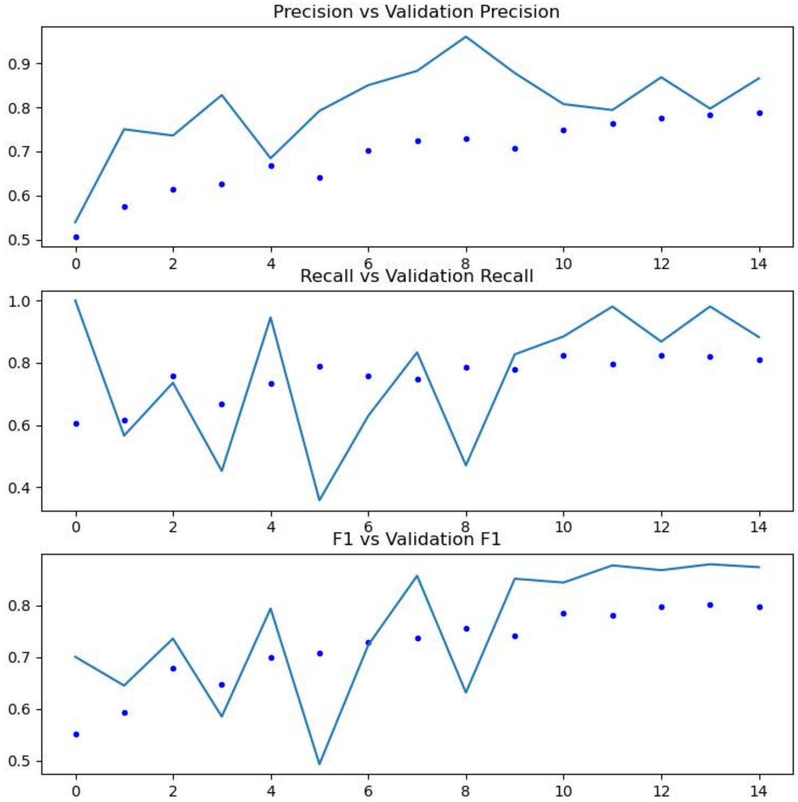



```
Found 376 images belonging to 2 classes.
Found 50 images belonging to 2 classes.
Found 112 images belonging to 2 classes.
```

Model

```
[339]: model2 = models.Sequential()
model2.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 3)))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Conv2D(128, (3, 3), activation='relu'))
model2.add(layers.Conv2D(256, (3, 3), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Conv2D(512, (3, 3), activation='relu'))
model2.add(layers.Conv2D(256, (3, 3), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Flatten())
model2.add(layers.Dropout(0.5))
model2.add(layers.Dense(256, activation='relu'))
model2.add(layers.Dense(1, activation='sigmoid'))

model2.summary()
```

— validation
.... train



Acc = 80 %

```
model2.evaluate(test_generator)

4/4 [==============================] - 2s 334ms/step - loss: 0.4221 - accuracy: 0.8000 - precision_
5: 0.8000 - recall_7: 0.8000

[0.42211291193962097, 0.800000011920929, 0.800000011920929, 0.800000011920929]
```

Augment fundus images

[2]:
```
import Augmentor
```

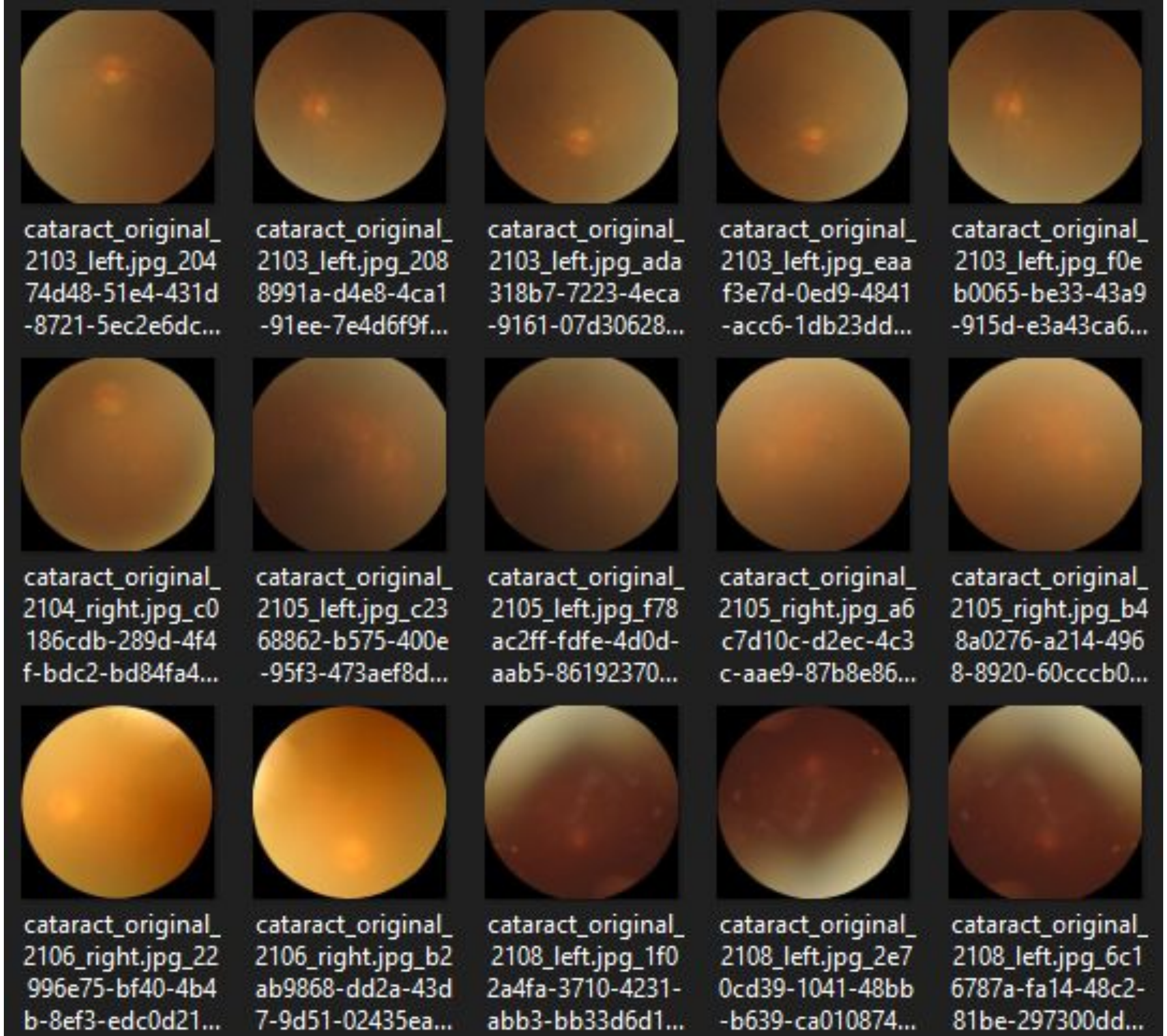
[9]:
```
img_dir_path1 = "./dataset256_multi/train/cataract"
img_dir_path2 = "./dataset256_multi/train/myopia"
img_dir_path3 = "./dataset256_multi/train/age_macular"

p = Augmentor.Pipeline(img_dir_path3)
```

```
Initialised with 186 image(s) found.
Output directory set to ./dataset256_multi/train/age_macular\output.
```

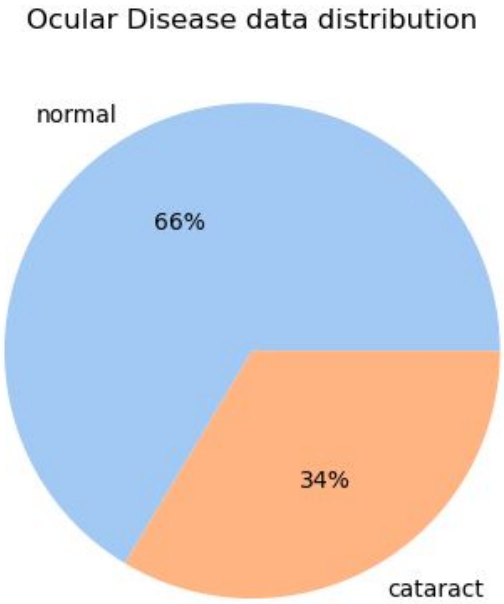
[10]:
```
p.rotate(probability=1, max_left_rotation=7, max_right_rotation=7)
# p.zoom(probability=0.3, min_factor=0.95, max_factor=1.05)
p.flip_left_right(probability=0.7)
p.flip_top_bottom(probability=0.7)
p.rotate_random_90(probability=0.7)
p.zoom(probability=0.5, min_factor=0.9, max_factor=1.1)
```

[11]:
```
p.sample(186*4)
```

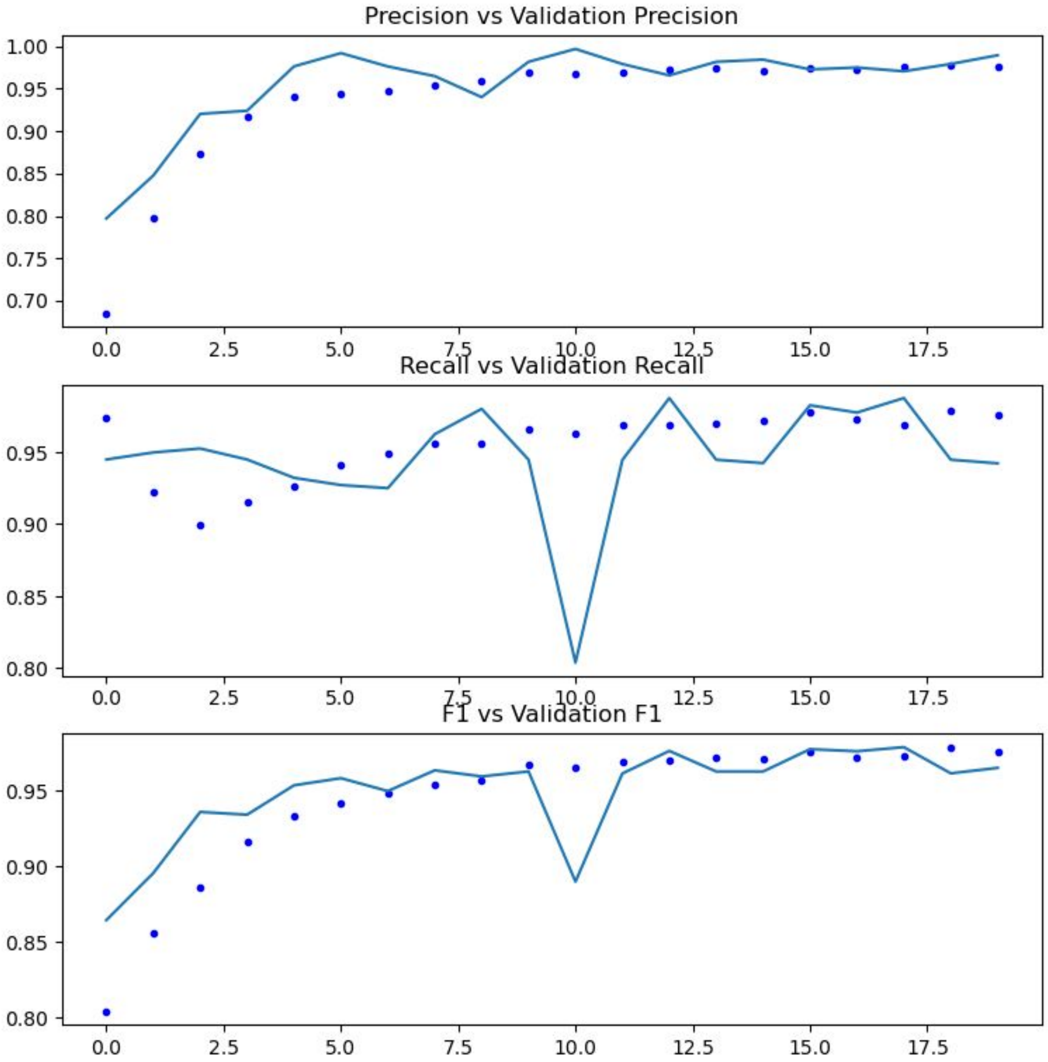
```
Processing <PIL.Image.Image image mode=RGB size=256x256 at 0x207E9FCCC40>: 100%|██████████| 744/744 [00:02
<00:00, 291.77 Samples/s]
```


cataract_original_2103_left.jpg_20474d48-51e4-431d-8721-5ec2e6dc...
cataract_original_2103_left.jpg_2088991a-d4e8-4ca1-91ee-7e4d6f9f...
cataract_original_2103_left.jpg_ada318b7-7223-4eca-9161-07d30628...
cataract_original_2103_left.jpg_eaaf3e7d-0ed9-4841-acc6-1db23dd...
cataract_original_2103_left.jpg_f0eb0065-be33-43a9-915d-e3a43ca6...
cataract_original_2104_right.jpg_c0186cdb-289d-4f4f-bdc2-bd84fa4...
cataract_original_2105_left.jpg_c2368862-b575-400e-95f3-473aef8d...
cataract_original_2105_left.jpg_f78ac2ff-fdfe-4d0d-aab5-86192370...
cataract_original_2105_right.jpg_a6c7d10c-d2ec-4c3c-aae9-87b8e86...
cataract_original_2105_right.jpg_b48a0276-a214-4968-8920-60cccb0...
cataract_original_2106_right.jpg_22996e75-bf40-4b4b-8ef3-edc0d21...
cataract_original_2106_right.jpg_b2ab9868-dd2a-43d7-9d51-02435ea...
cataract_original_2108_left.jpg_1f02a4fa-3710-4231-abb3-bb33d6d1...
cataract_original_2108_left.jpg_2e70cd39-1041-48bb-b639-ca010874...
cataract_original_2108_left.jpg_6c16787a-fa14-48c2-81be-297300dd...

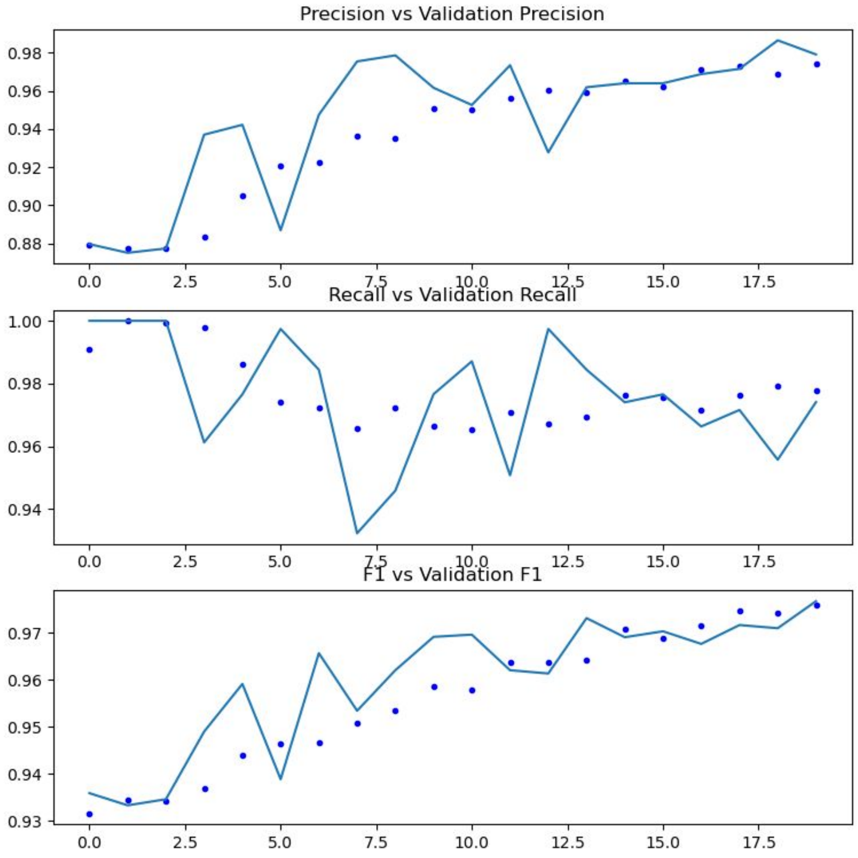
Wyniki po augmentacji



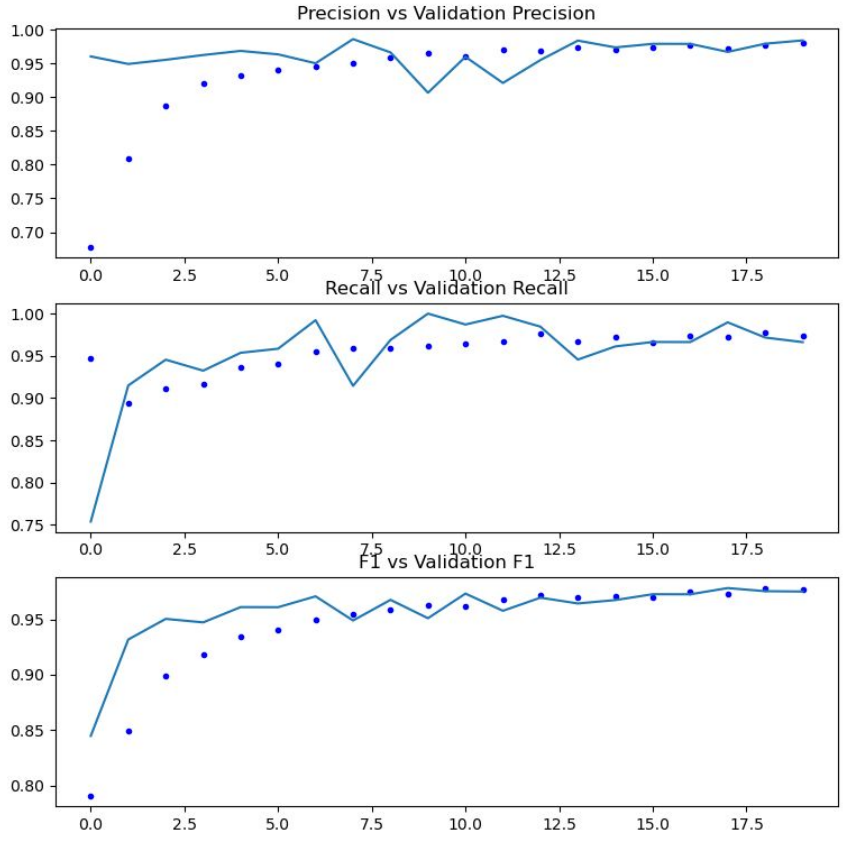

```
Found 2019 images belonging to 2 classes.
Found 261 images belonging to 2 classes.
Found 605 images belonging to 2 classes.
```



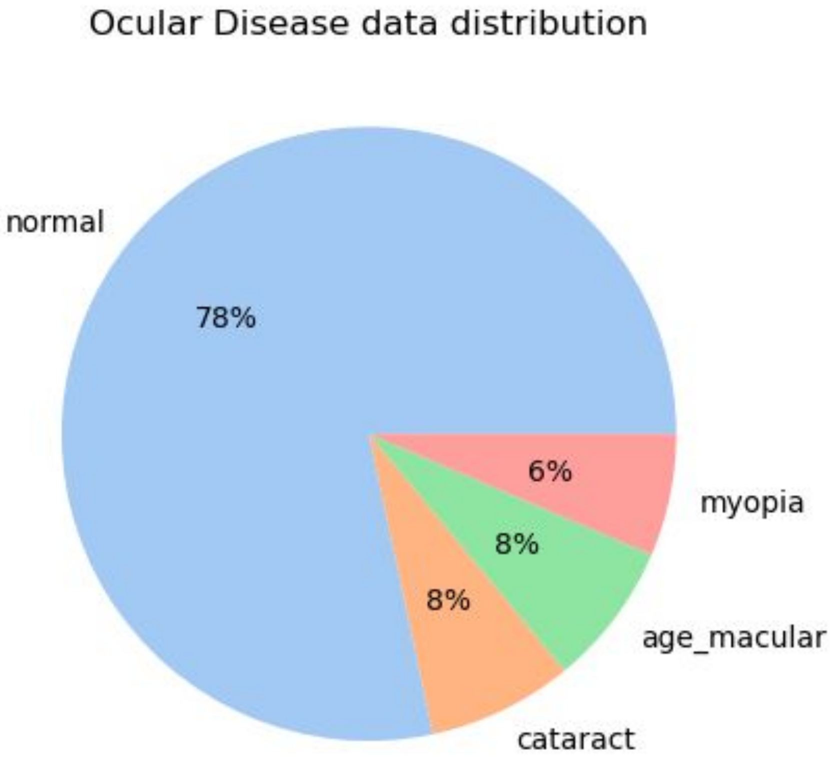
Bez augmentacji (269 ‘cataract’, 2000 ‘normal’)



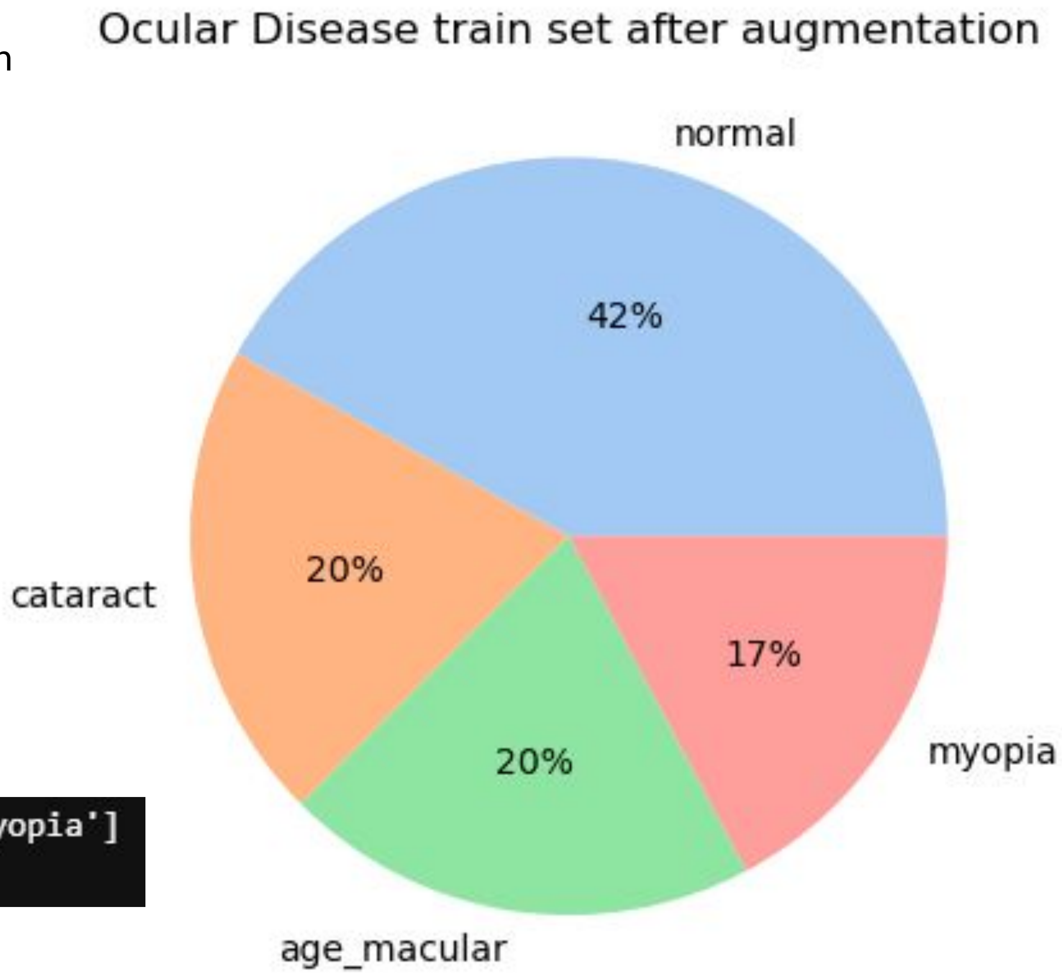
Po augmentacji(800 ‘cataract’, 2000 ‘normal’)



Powrót do klasyfikacji wieloklasowej



Zbiór walidacyjny i testowy nie zawierają danych augmentowanych.



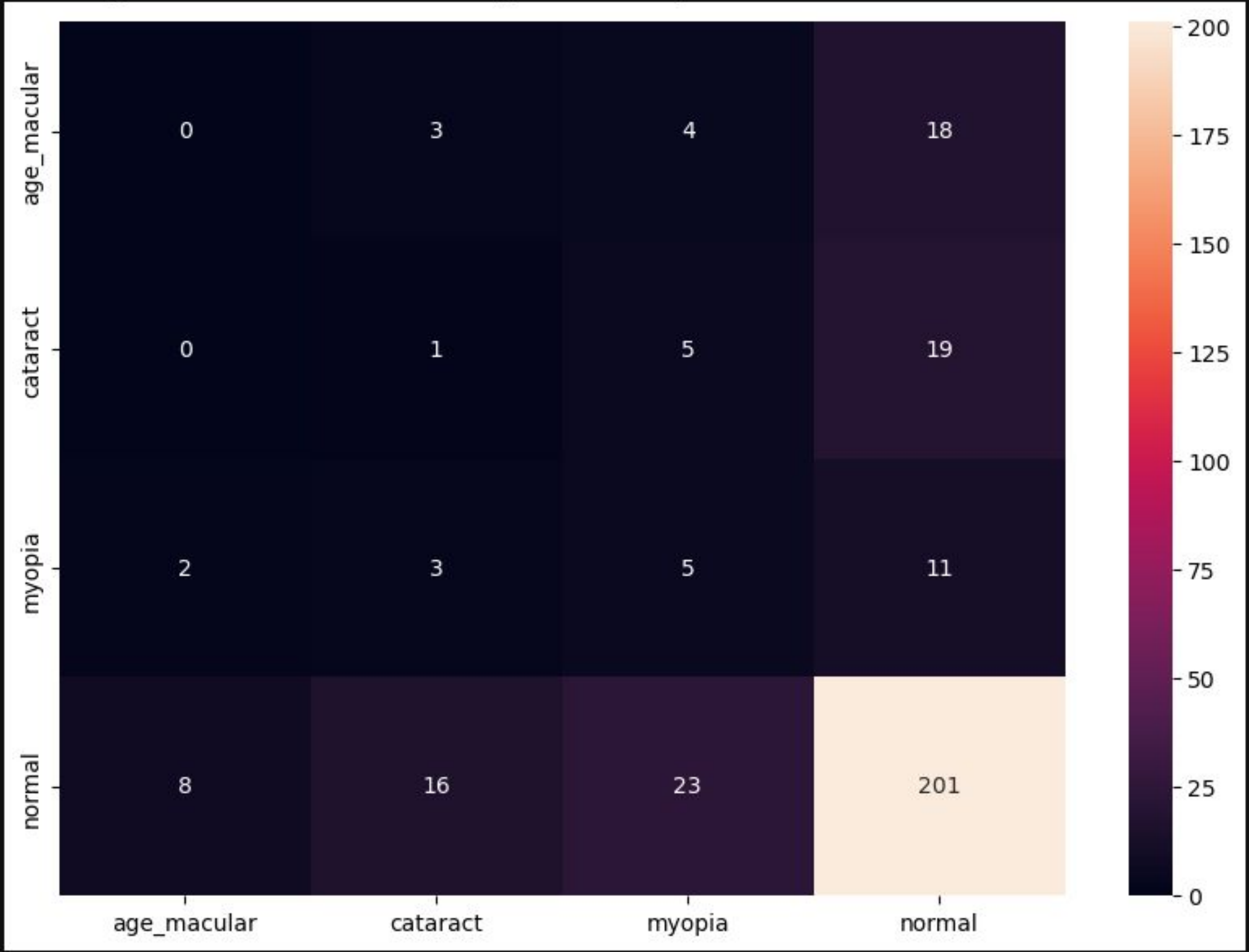

```
['normal', 'cataract', 'age_macular', 'myopia']
[1926, 940, 930, 790]
```

[111]:
```python
model2 = models.Sequential()
model2.add(layers.Conv2D(512, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 3)))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Conv2D(512, (3, 3), activation='relu'))
model2.add(layers.Conv2D(512, (3, 3), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Conv2D(256, (3, 3), activation='relu'))
model2.add(layers.MaxPooling2D((2, 2), padding='same'))
model2.add(layers.Conv2D(256, (3, 3), activation='relu'))
model2.add(layers.Flatten())
model2.add(layers.Dense(256, activation='relu'))
model2.add(layers.Dense(class_num, activation='softmax'))

model2.summary()
```

```
Model: "sequential_14"
```

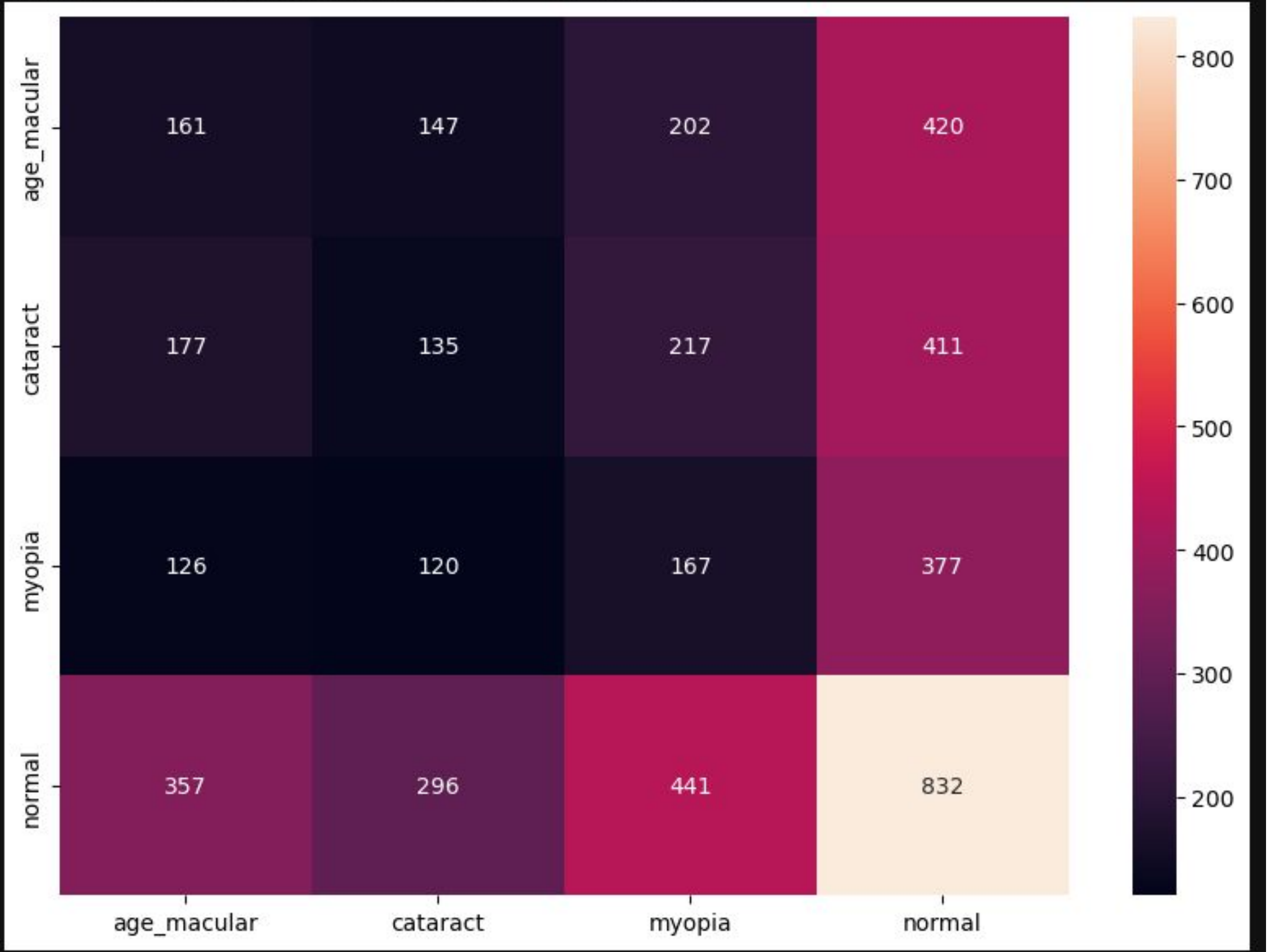
[116]:
```
plot_conf_matrix(model2, test_generator, classes)
```

```
160/160 [==============================] - 1s 7ms/step
```

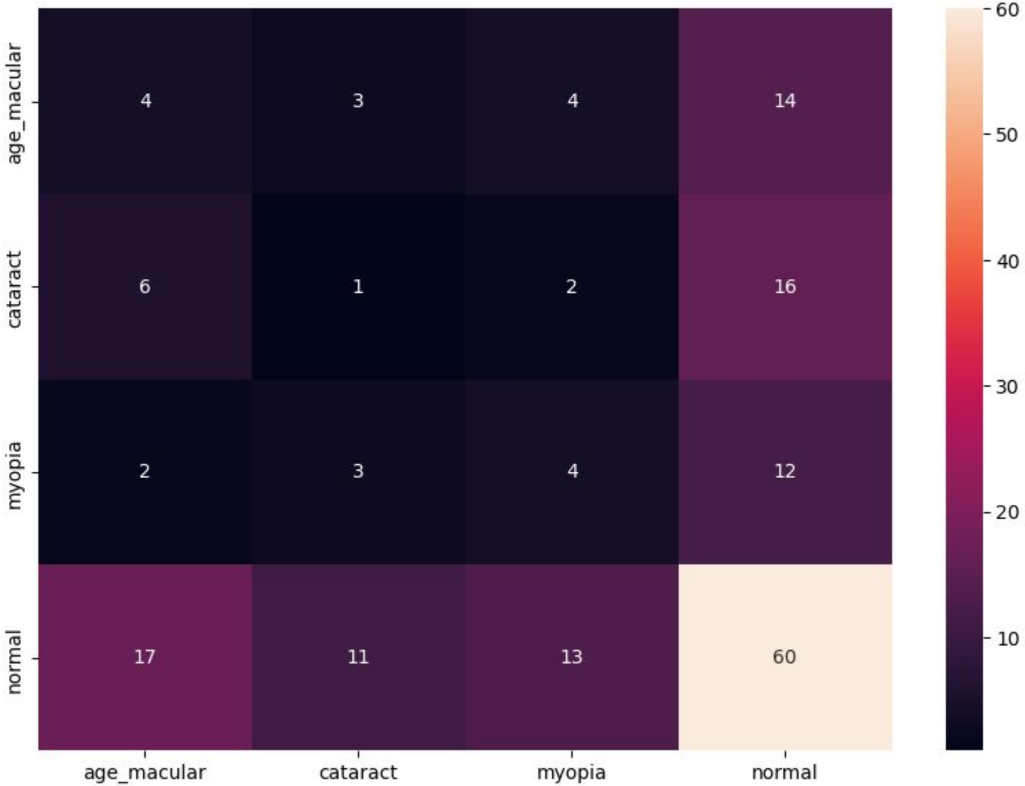


```
[96]: plot_conf_matrix(model2, train_generator, classes)
```

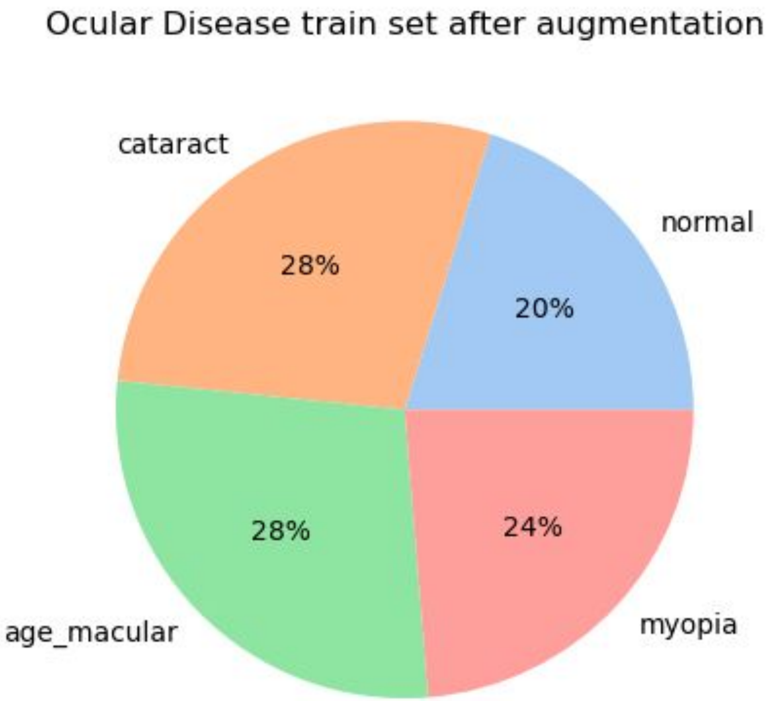
```
918/918 [==============================] - 11s 12ms/step
```



Brak większych rezultatów dla zbioru testowego

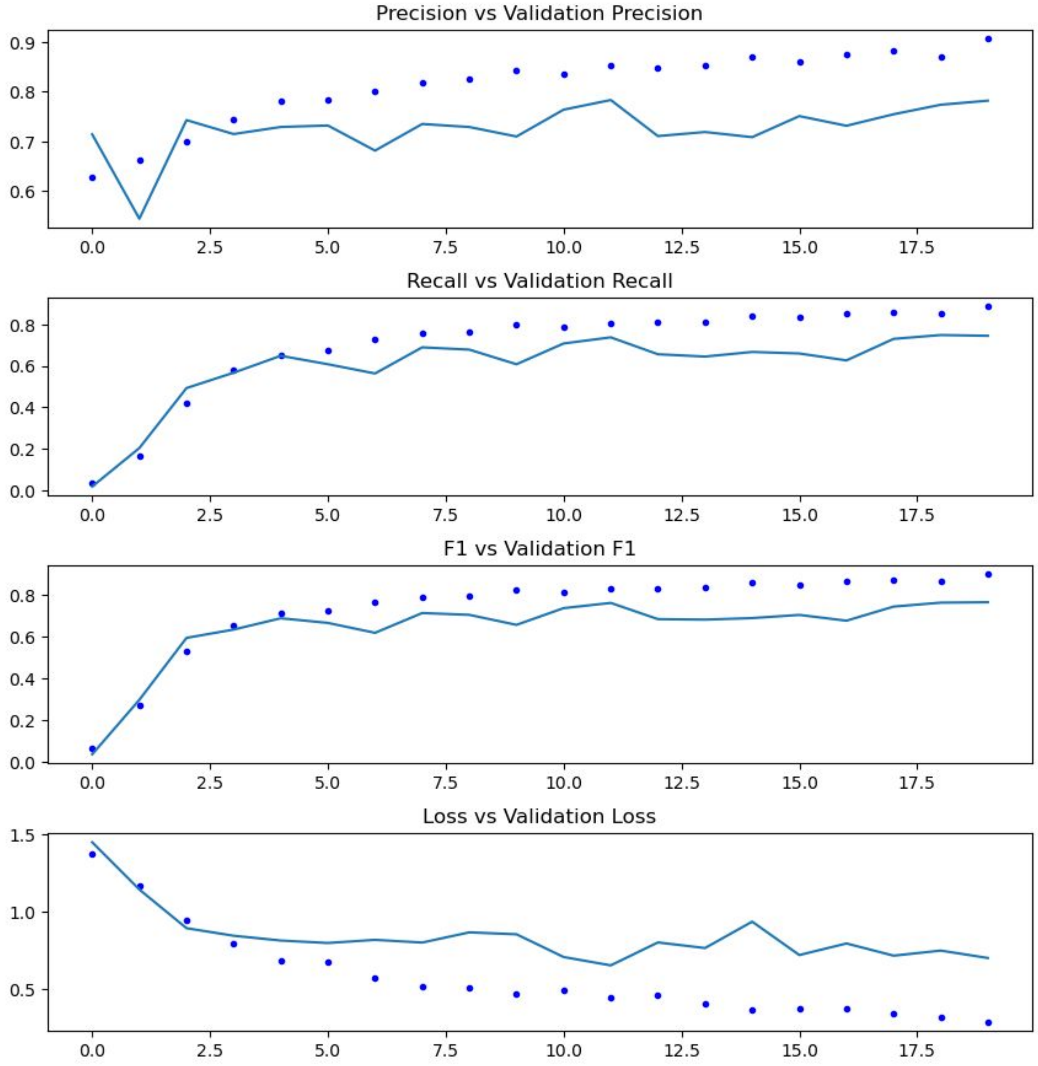


Zmniejszenie ilości zdjęć klasy ‘normal’ w zbiorze treningowym (undersampling)



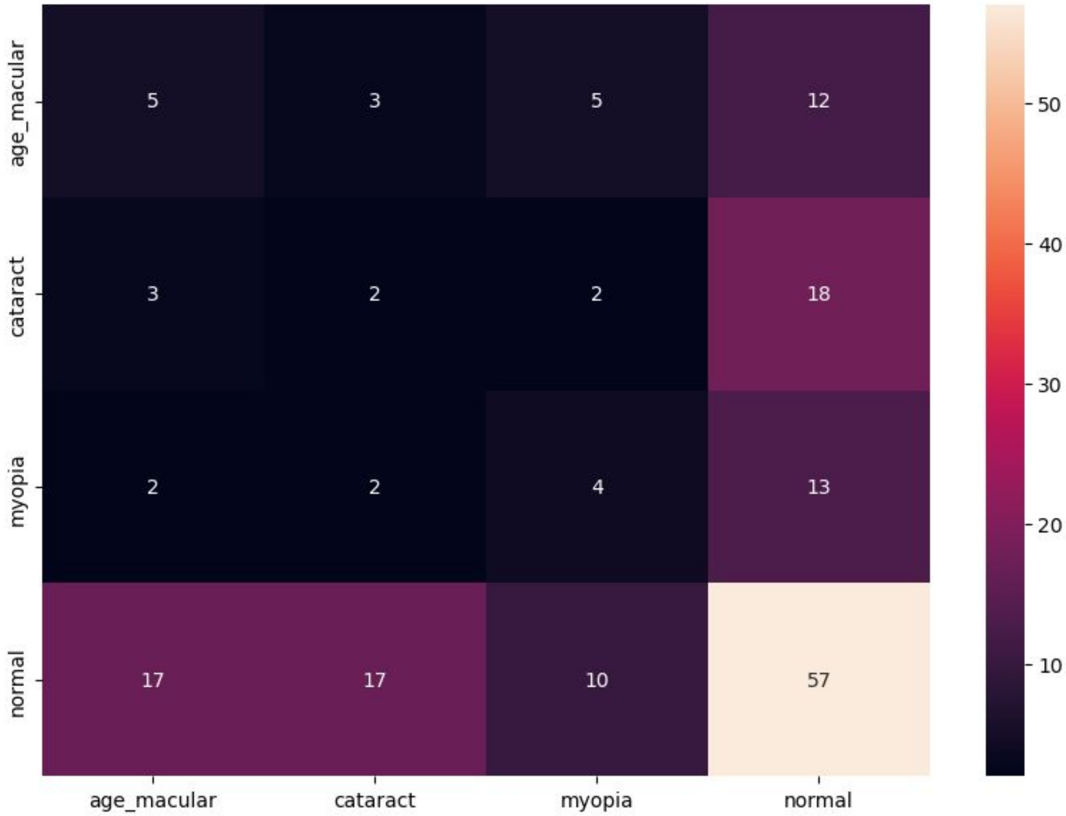
Kolejny model

```
model5 = models.Sequential()
model5.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 3)))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Conv2D(256, (3, 3), activation='relu'))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Conv2D(512, (3, 3), activation='relu'))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Conv2D(256, (3, 3), activation='relu'))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Conv2D(256, (3, 3), activation='relu'))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Conv2D(128, (3, 3), activation='relu'))
model5.add(layers.MaxPooling2D((2, 2), padding='same'))
model5.add(layers.Flatten())
model5.add(layers.Dense(128, activation='relu'))
model5.add(layers.Dense(class_num, activation='softmax'))

model5.summary()
```

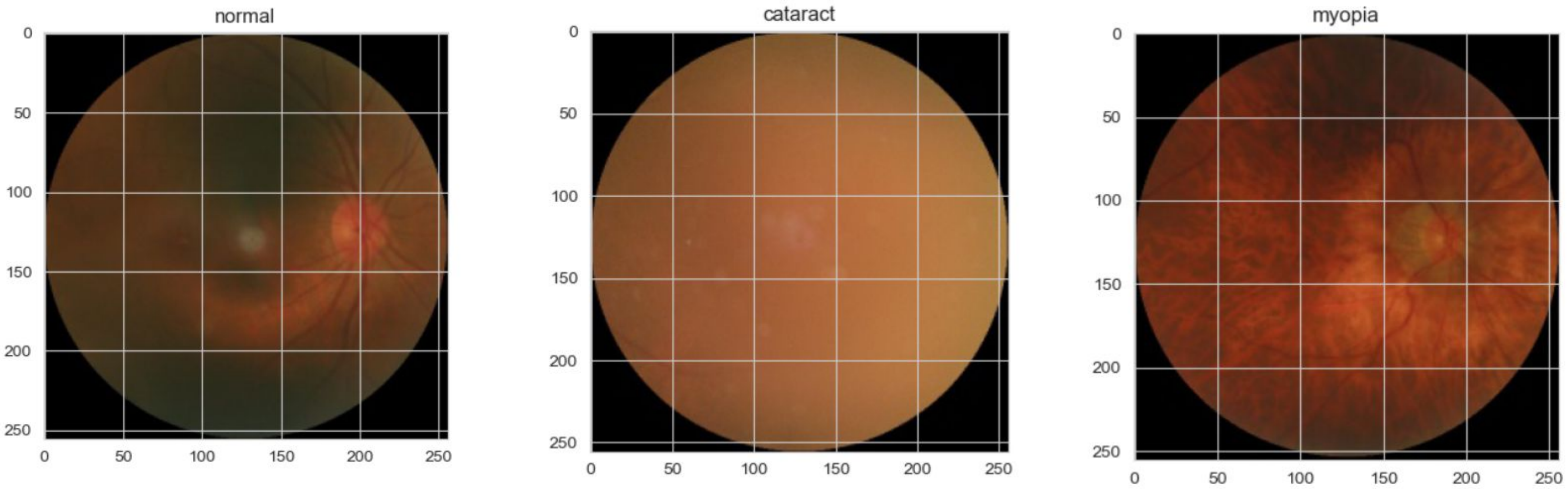

Precision vs Validation Precision
0.6
0.7
0.8
0.9
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Recall vs Validation Recall
0.0
0.2
0.4
0.6
0.8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
F1 vs Validation F1
0.0
0.2
0.4
0.6
0.8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Loss vs Validation Loss
0.5
1.0
1.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

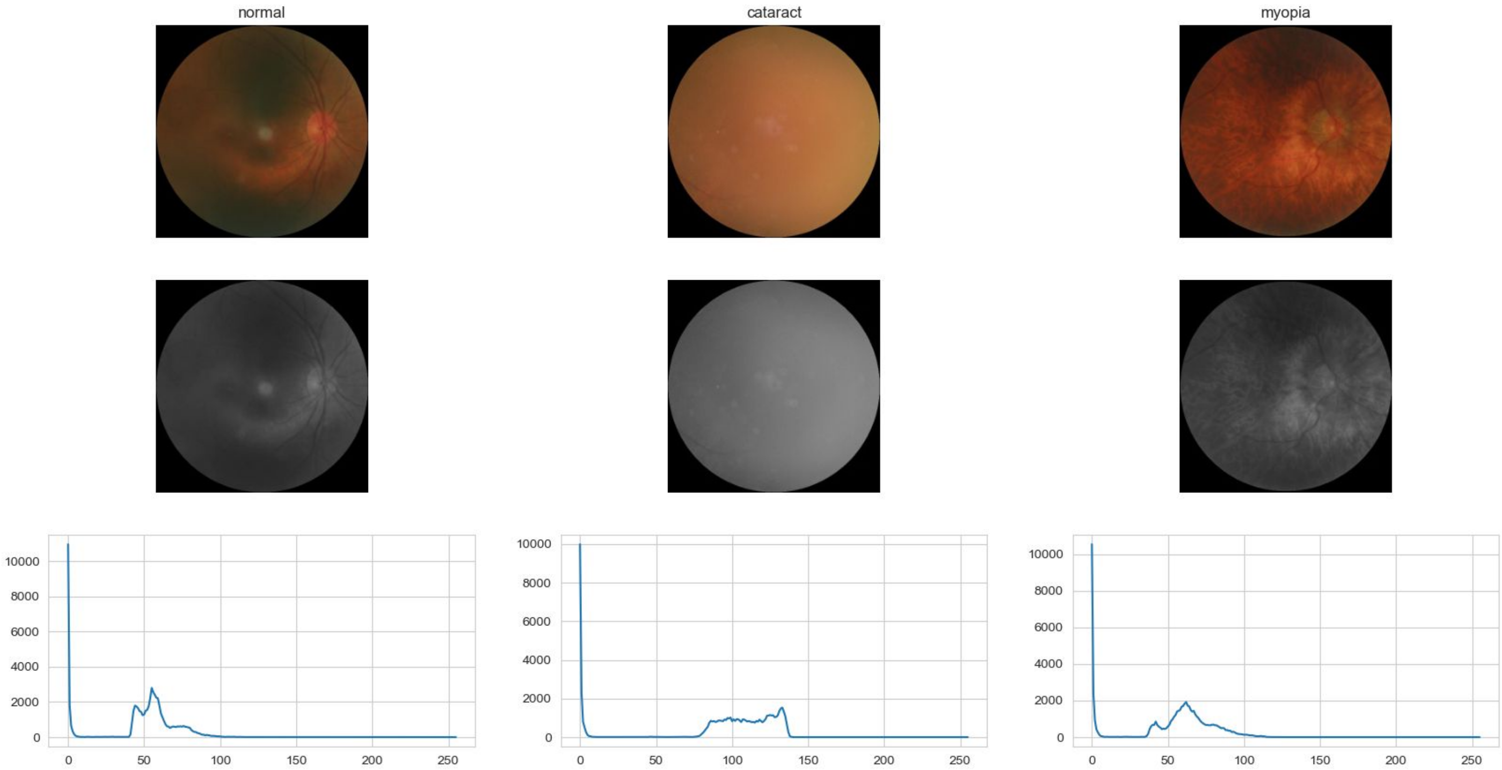
```
model5.evaluate(test_generator)
```

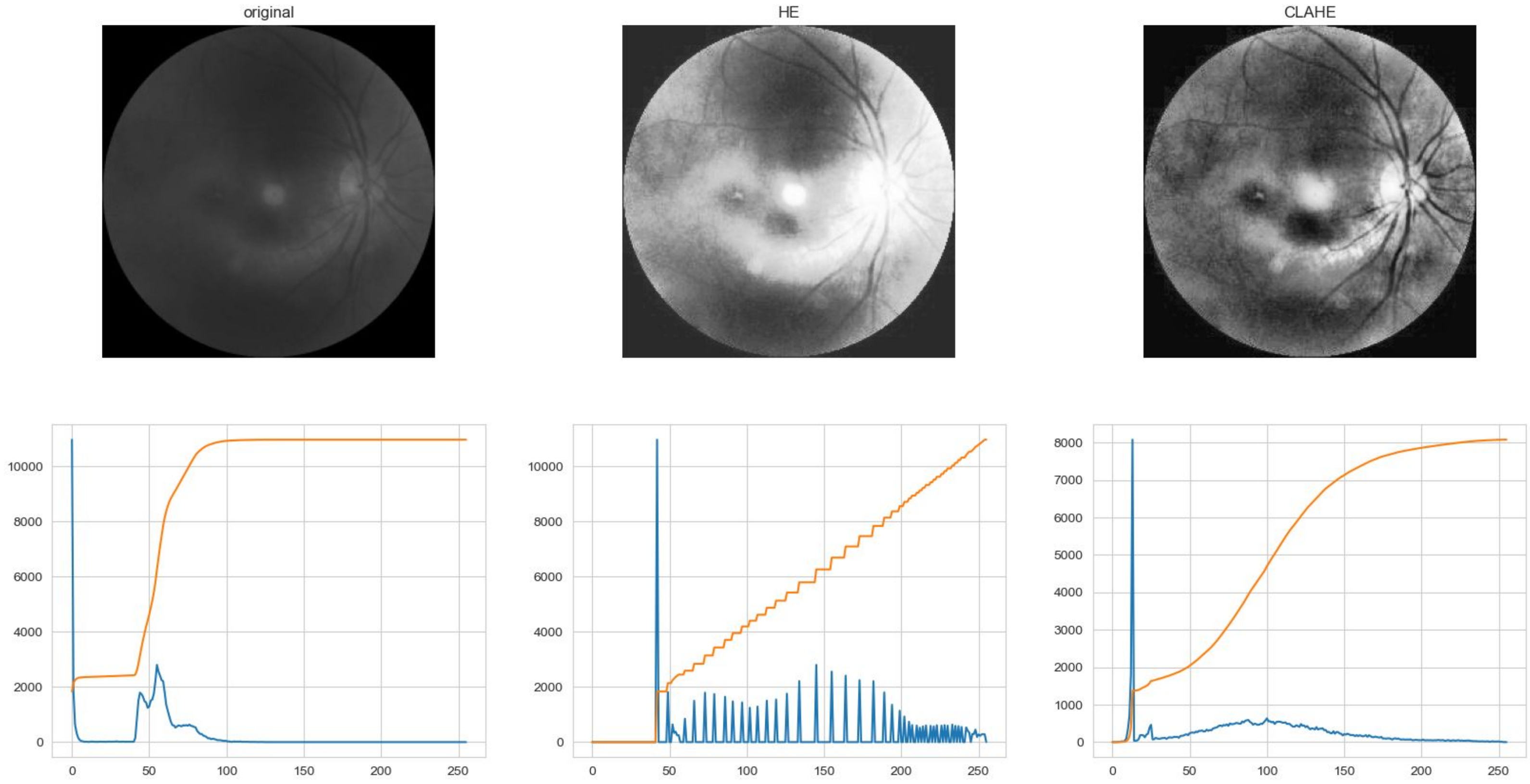
```
12/12 [==============================] - 1s 75ms/step - loss: 0.6237 - precision_18: 0.7784 - recall_18: 0.7558
```



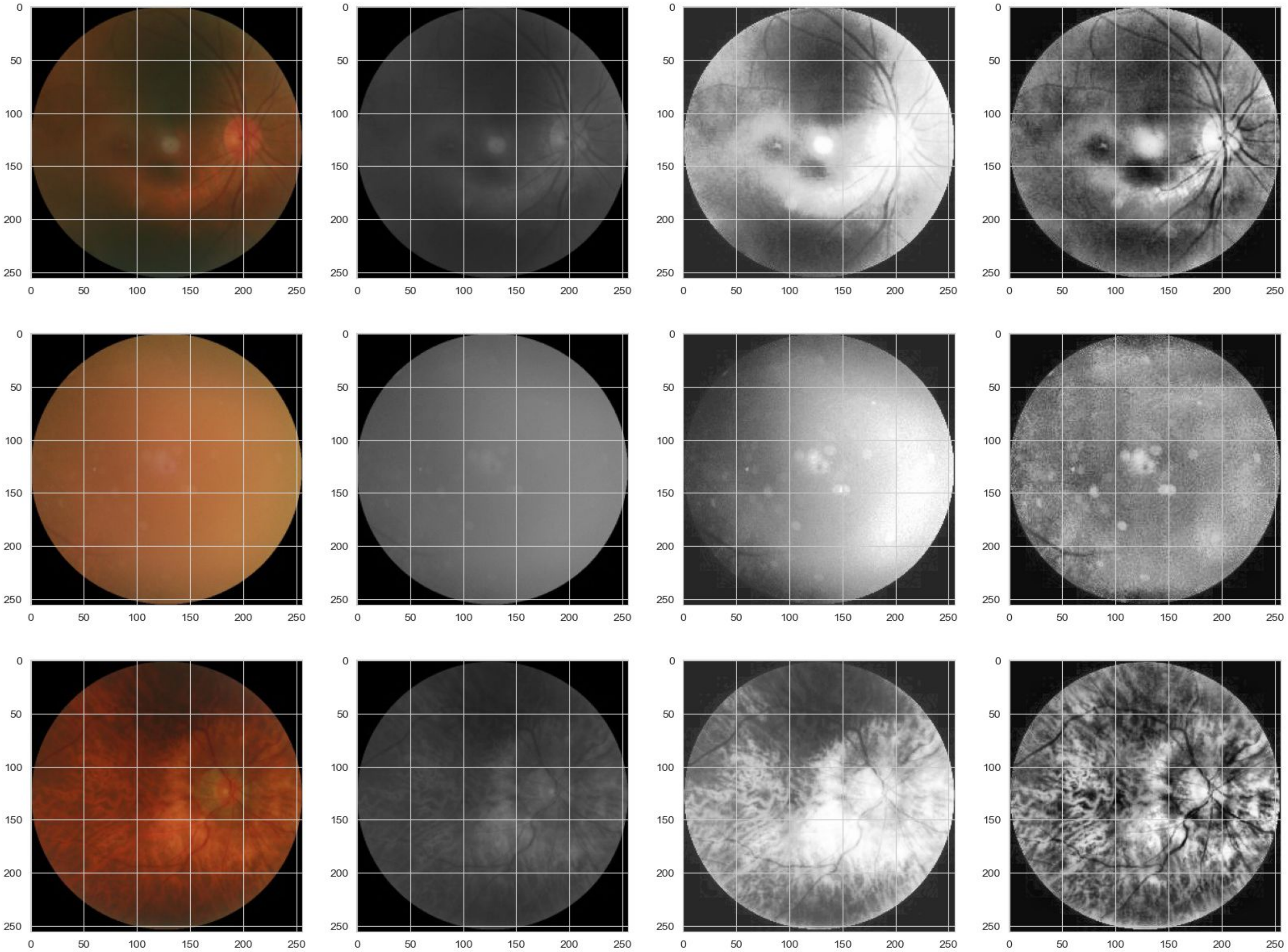
Przetwarzanie obrazów




normal
cataract
myopia
10000
8000
6000
4000
2000
0
0
50
100
150
200
250



original
HE
CLAHE
0
2000
4000
6000
8000
10000
0
50
100
150
200
250
1000
3000
5000
7000
8000



Model

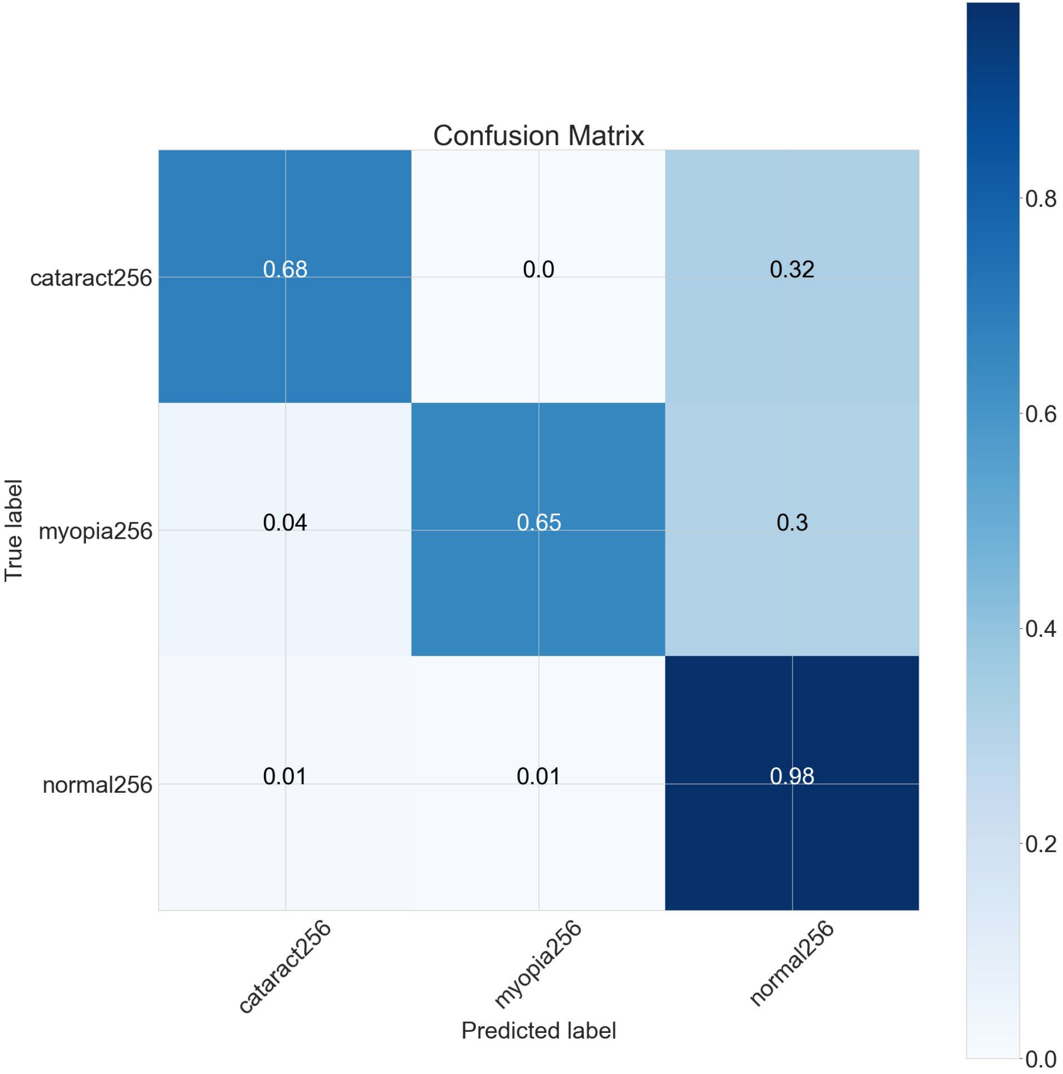
```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))

model1.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
               optimizer = tf.keras.optimizers.Adam(),
               metrics = [tf.keras.metrics.Precision(), tf.keras.metrics.Recall(), 'accuracy'])
```

```
SET 0 - without oversampling
Found 2272 images belonging to 3 classes.
Found 327 images belonging to 3 classes.
Found 648 images belonging to 3 classes.
```

```
Classification Report
              precision    recall  f1-score   support

 cataract256       0.83      0.68      0.75        28
   myopia256       0.88      0.65      0.75        23
   normal256       0.94      0.98      0.96       276

    accuracy                           0.93       327
   macro avg       0.88      0.77      0.82       327
weighted avg       0.93      0.93      0.93       327
```

Test / Train set:

- 276 / 1926 normal
- 28 / 188 cataract
- 23 / 158 myopia



```
SET 1 - with oversampling
Found 2865 images belonging to 3 classes.
Found 45 images belonging to 3 classes.
Found 90 images belonging to 3 classes.
```

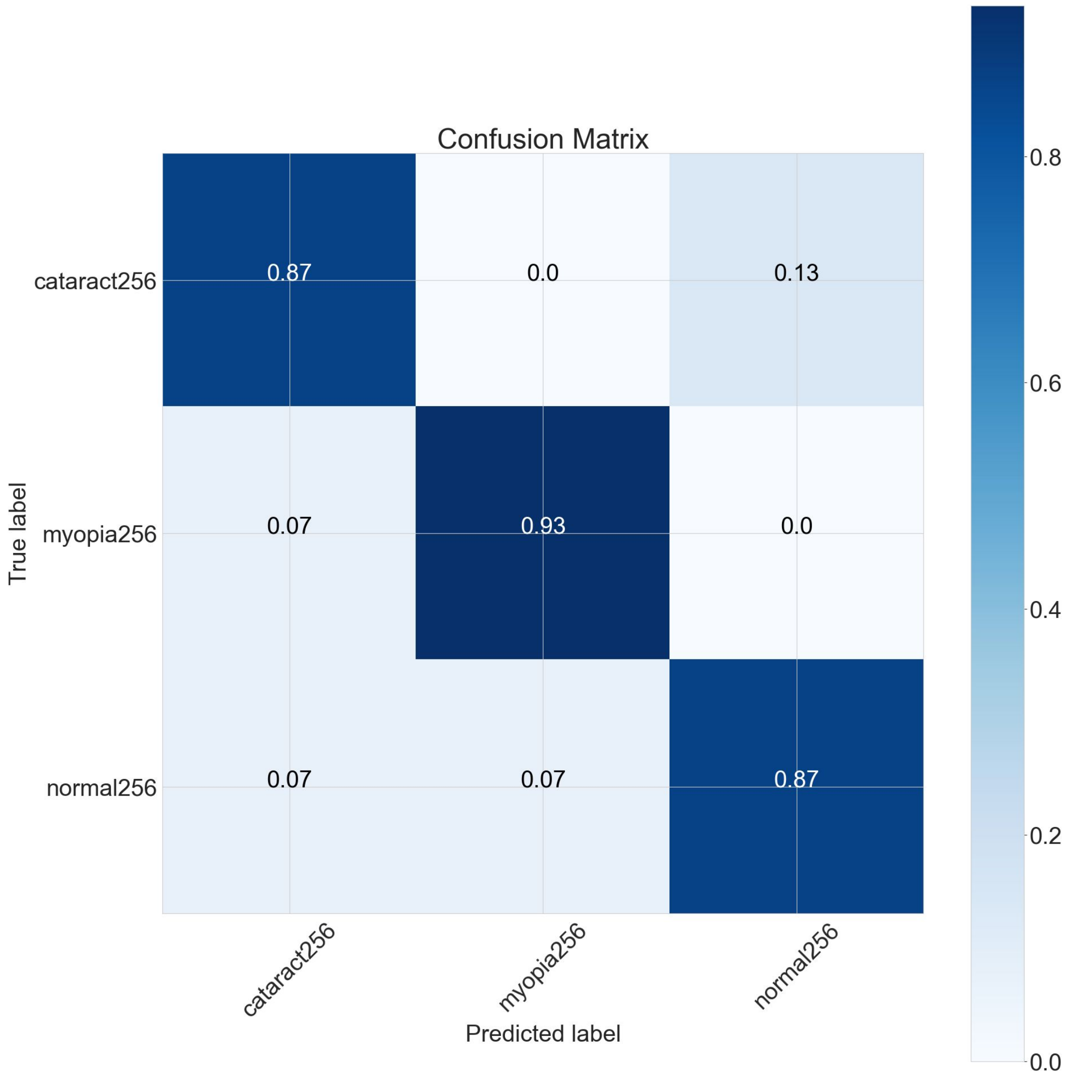
```
Classification Report
              precision    recall  f1-score   support

 cataract256       0.87      0.87      0.87        15
   myopia256       0.93      0.93      0.93        15
   normal256       0.87      0.87      0.87        15

    accuracy                           0.89        45
   macro avg       0.89      0.89      0.89        45
weighted avg       0.89      0.89      0.89        45
```

Train:

- 955 images each class

Test set:

- 15 normal
- 15 cataract
- 15 myopia



Model 2

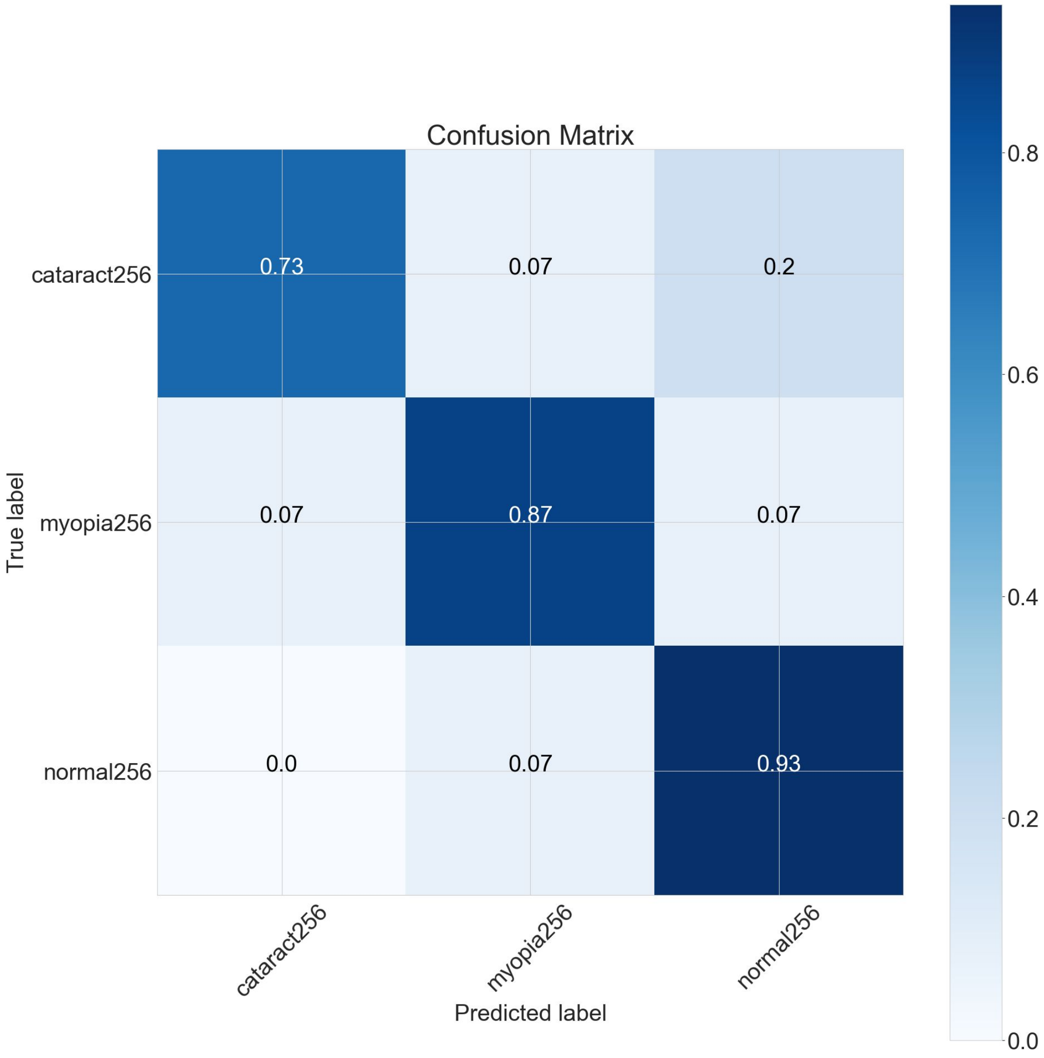
```
model = models.Sequential()
model.add(layers.Conv2D(64, (3, 3), activation='relu', input_shape=(img_rows, img_cols, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(512, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(256, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(3, activation='softmax'))
model3.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
               optimizer = tf.keras.optimizers.Adam(),
               metrics = [tf.keras.metrics.Precision(), tf.keras.metrics.Recall(), 'accuracy'])
```

```
SET 1 - with oversampling
Found 2865 images belonging to 3 classes.
Found 45 images belonging to 3 classes.
Found 90 images belonging to 3 classes.
```

```
Classification Report
              precision    recall  f1-score   support

 cataract256       0.87      0.87      0.87        15
   myopia256       0.93      0.93      0.93        15
   normal256       0.87      0.87      0.87        15

    accuracy                           0.89        45
   macro avg       0.89      0.89      0.89        45
weighted avg       0.89      0.89      0.89        45
```

Train:
- 955 images each class

Test set:
- 15 normal
- 15 cataract
- 15 myopia



```
SET 2 - with oversampling and increased images in test & validation sets
Found 2760 images belonging to 3 classes.
Found 120 images belonging to 3 classes.
Found 120 images belonging to 3 classes.
```

Classification Report

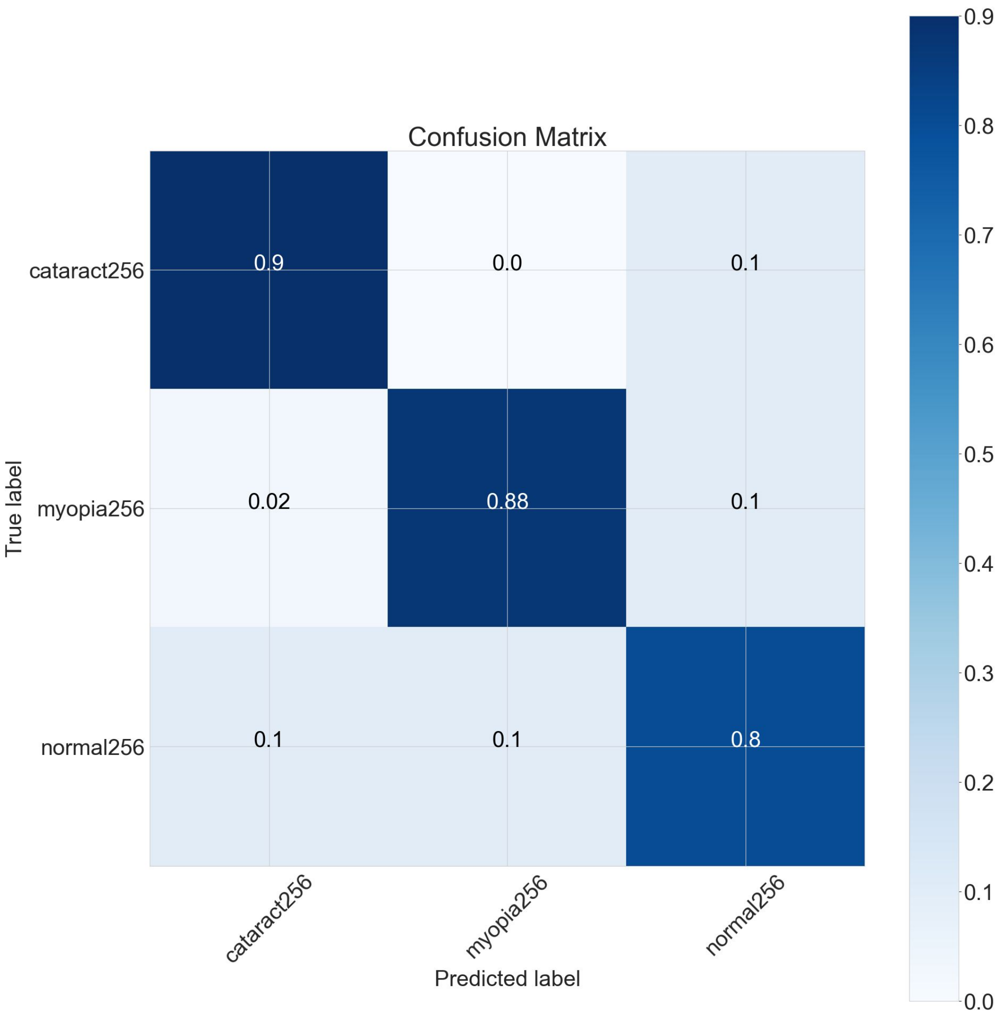
	precision	recall	f1-score	support
cataract256	0.88	0.90	0.89	40
myopia256	0.90	0.88	0.89	40
normal256	0.80	0.80	0.80	40
accuracy			0.86	120
macro avg	0.86	0.86	0.86	120
weighted avg	0.86	0.86	0.86	120

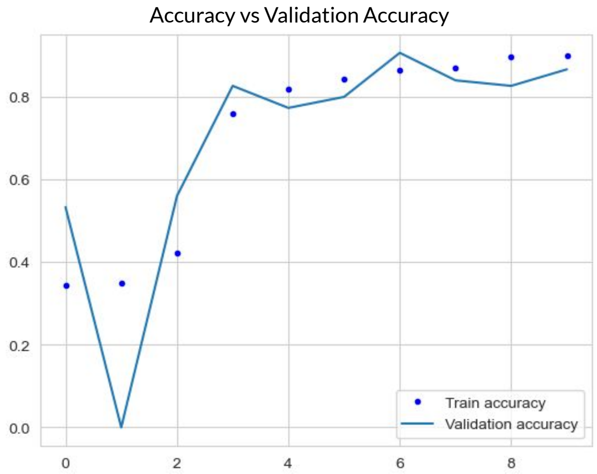
Train:

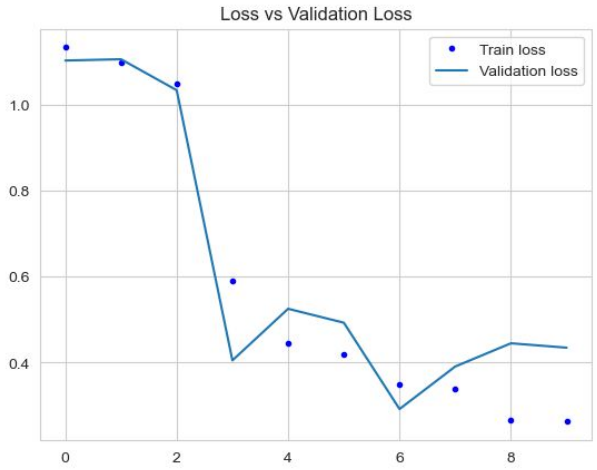
- 920 images each class

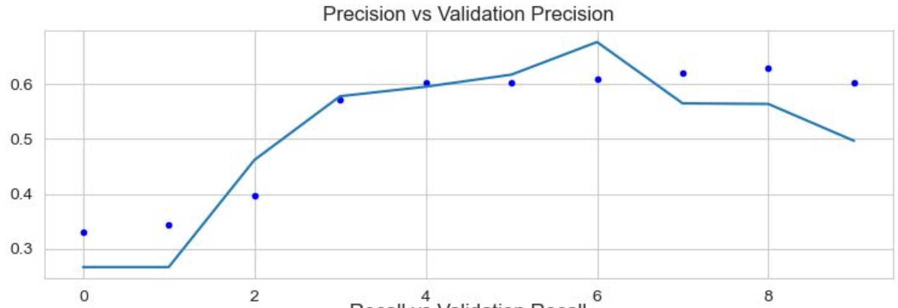
Test set:

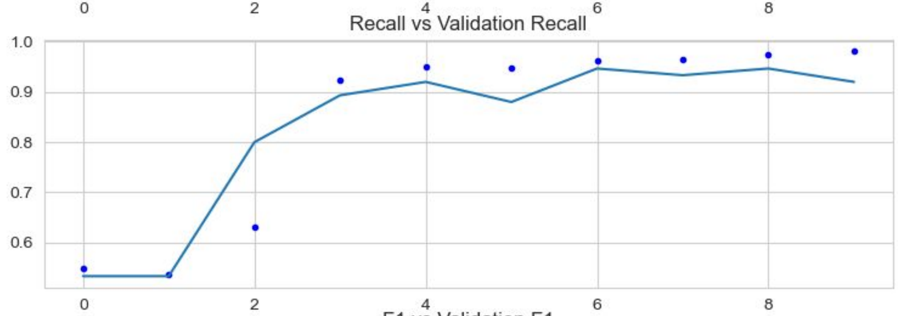
- 40 normal
- 40 cataract
- 40 myopia

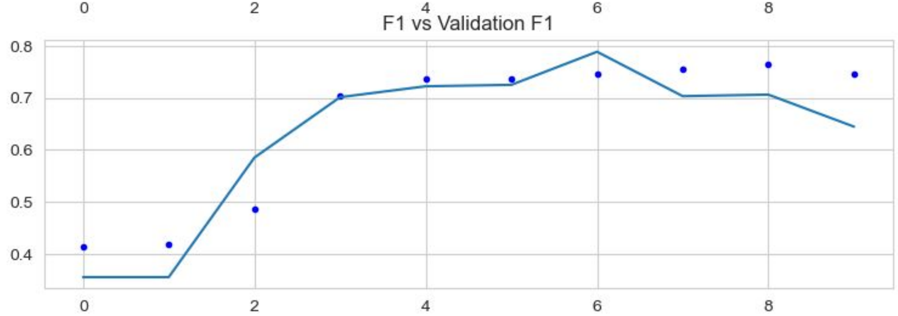



Accuracy vs Validation Accuracy
Train accuracy
Validation accuracy


Loss vs Validation Loss
Train loss
Validation loss


Precision vs Validation Precision


Recall vs Validation Recall


F1 vs Validation F1

Porównanie z RGB

Train:

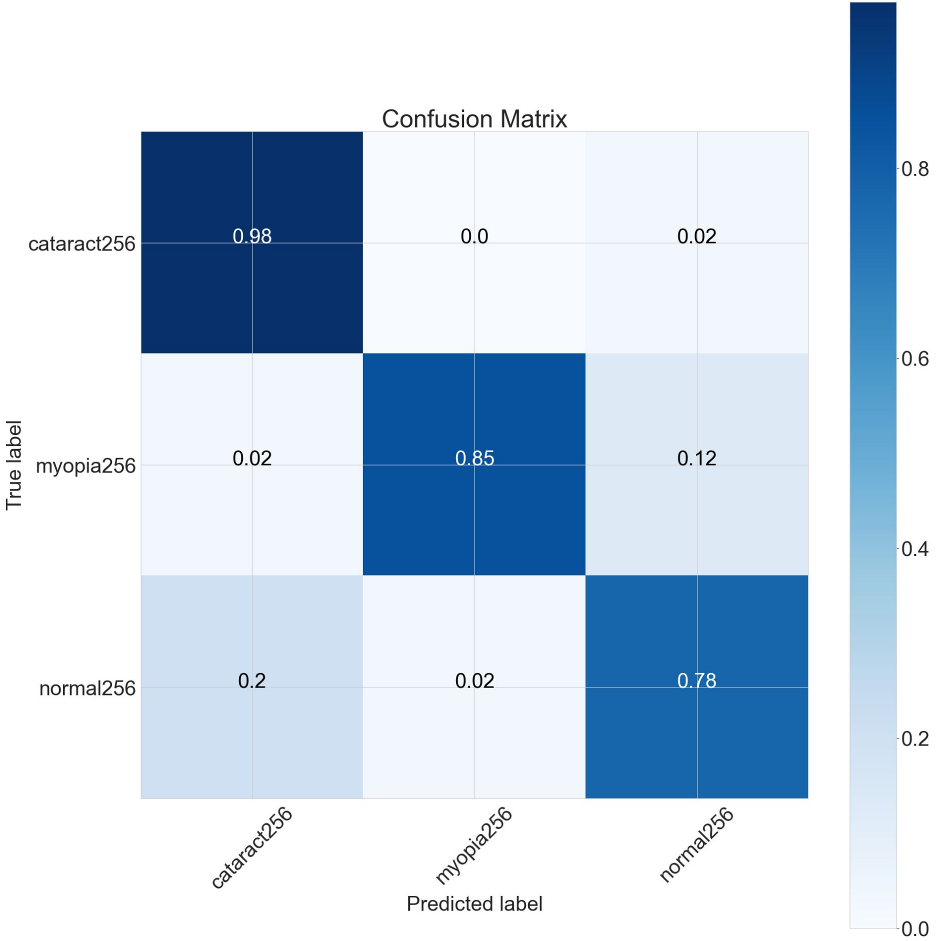
- 920 images each class

Test set:

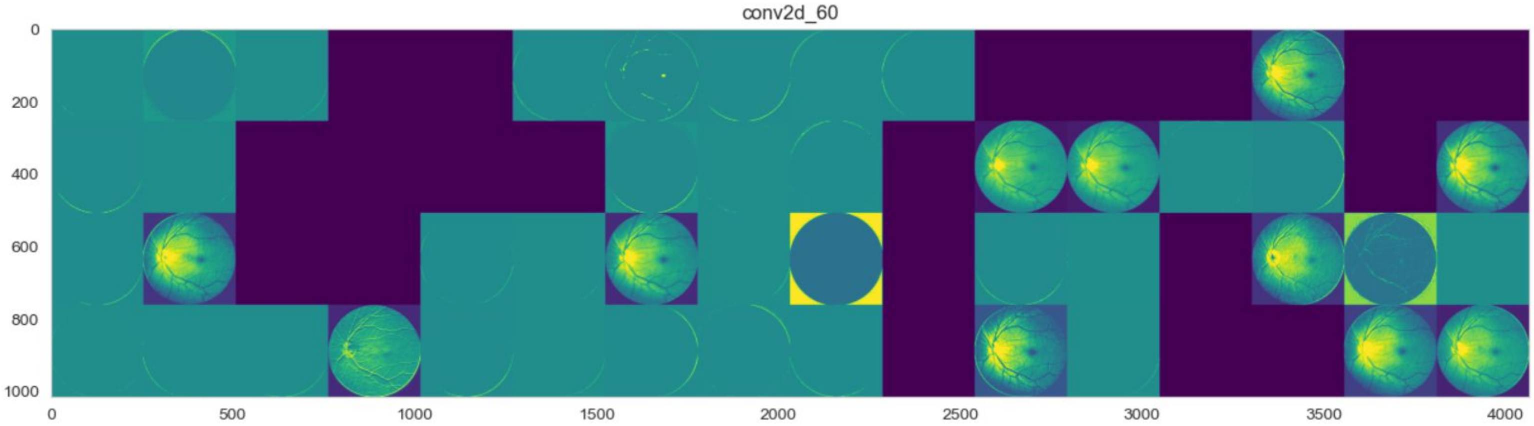
- 40 normal
- 40 cataract
- 40 myopia

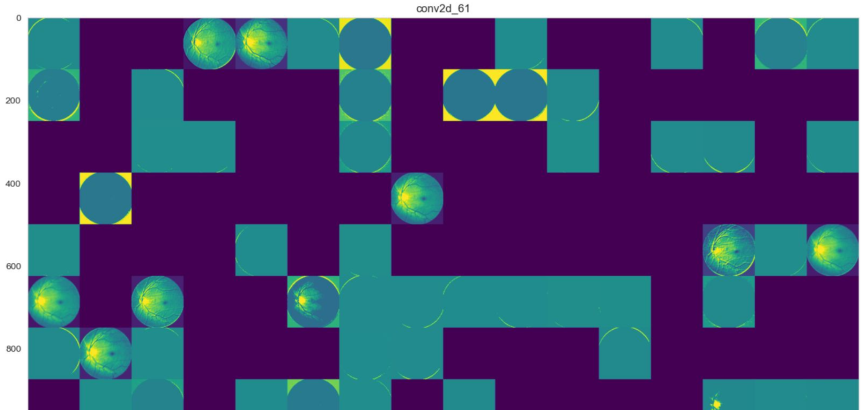
Classification Report

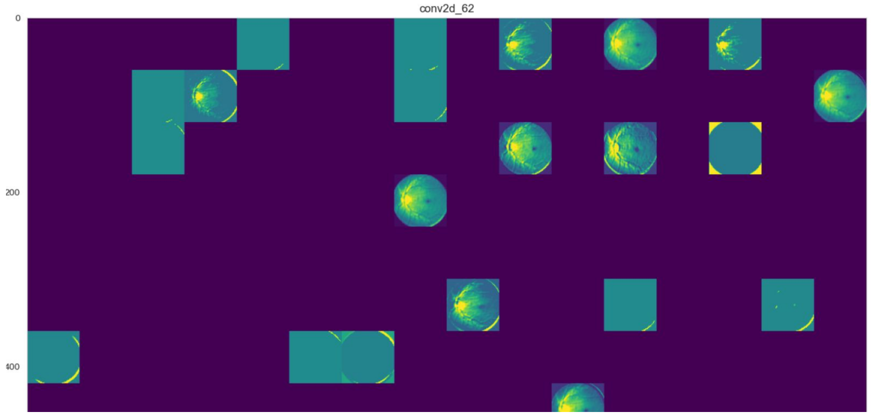
	precision	recall	f1-score	support
cataract256	0.81	0.97	0.89	40
myopia256	0.97	0.85	0.91	40
normal256	0.84	0.78	0.81	40
accuracy			0.87	120
macro avg	0.87	0.87	0.87	120
weighted avg	0.87	0.87	0.87	120



Filtry modelu

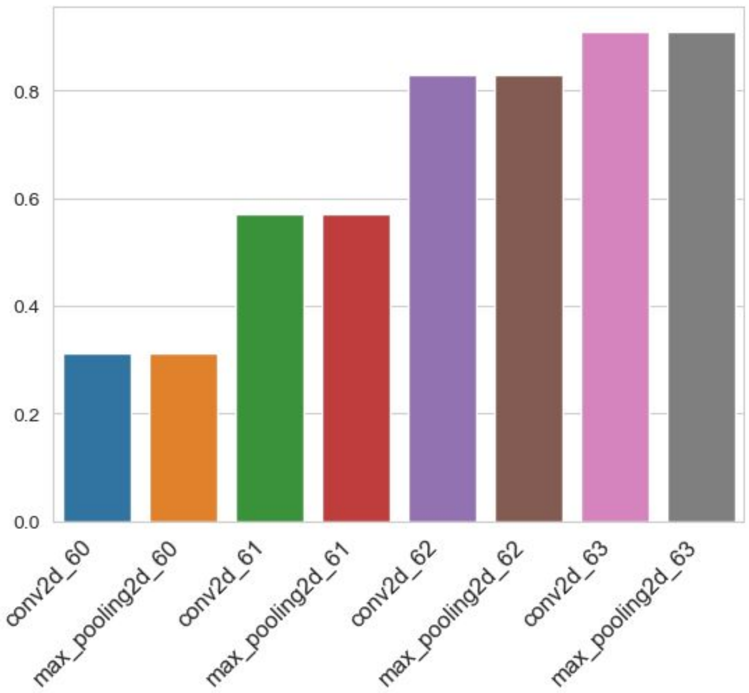






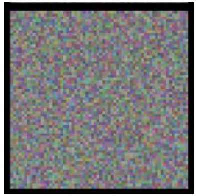
Filtry martwe

Potencjalnych filtrów martwych jest całkiem sporo, lecz aby potwierdzić, że filtry wykazujące brak wzorca są martwe potrzebne by było sprawdzenie czy sytuacja powtarza się nie dla jednego zdjęcia, a dla wielu, najlepiej różnorodnych zdjęć.



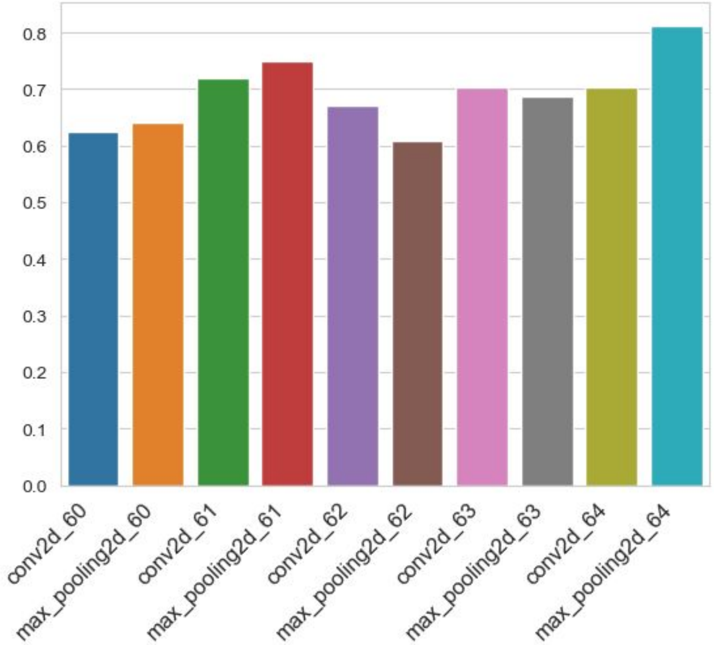
Procentowy udział potencjalnych filtrów martwych dla każdej z warstw

Występowanie szumu losowego



Ilość filtrów zawierających szum losowy w zbudowanym modelu jest na zaskakująco wysokim poziomie.

Kolejnymi krokami, które mogłyby pomóc w redukcji tak wielu filtrów szumu mogłaby być analiza zachowania sieci przy zastosowaniu różnych funkcji aktywacji w warstwach, lecz ograniczenia czasowe nie pozwoliły na sprawdzenie tego na ten moment.



Procentowy udział filtrów reprezentujących szum losowy dla każdej z warstw

Podsumowanie

Podczas projektu największym problemem była nierówny rozkład zdjęć pomiędzy klasy chorób oczu oraz różna jasność niezależnie od klasy do której zdjęcia należą.

Po kilku próbach uczenia na wszystkich klasach po paśmie porażek postanowiłem zacząć od klasyfikacji binarnej, która poszła o wiele lepiej.

Finalnie udało się dokonać stosunkowo dobrej klasyfikacji wieloklasowej dla klas myopia, normal, cataract.

Oczywiście sam model można poprawić na co wskazuje duża ilość filtrów szumu w ConvNet, lecz niestety z powodu intensywnej końcówki semestru zabrakło na to czasu.

Podsumowanie cd.

Na zajęciach pogłębiłem swoją wiedzę z zakresu obróbki danych do uczenia, tworzenia różnych architektur sieci neuronowych (głównie CNN) wraz z walidacją i analizą procesu ich uczenia.

Dodatkowo, zajęcia z pewnością pozwoliły rozeznać się jak szerokie i często wymagające są zagadnienia związane z sieciami neuronowymi. Z pewnością w najbliższym czasie skupię się bardziej na tych zagadnieniach, aby bardziej świadomie budować sieci i przy ich użyciu rozwiązywać coraz bardziej złożone problemy.

Sieci neuronowe jak i sztuczna inteligencja są bardzo ciekawym narzędziem, które pozwala na rozwiązanie wielu nietrywialnych problemów, a i ich użycie staje się coraz bardziej powszechne, co także motywuje do działania i zgłębiania informacji w tym zakresie.

The end