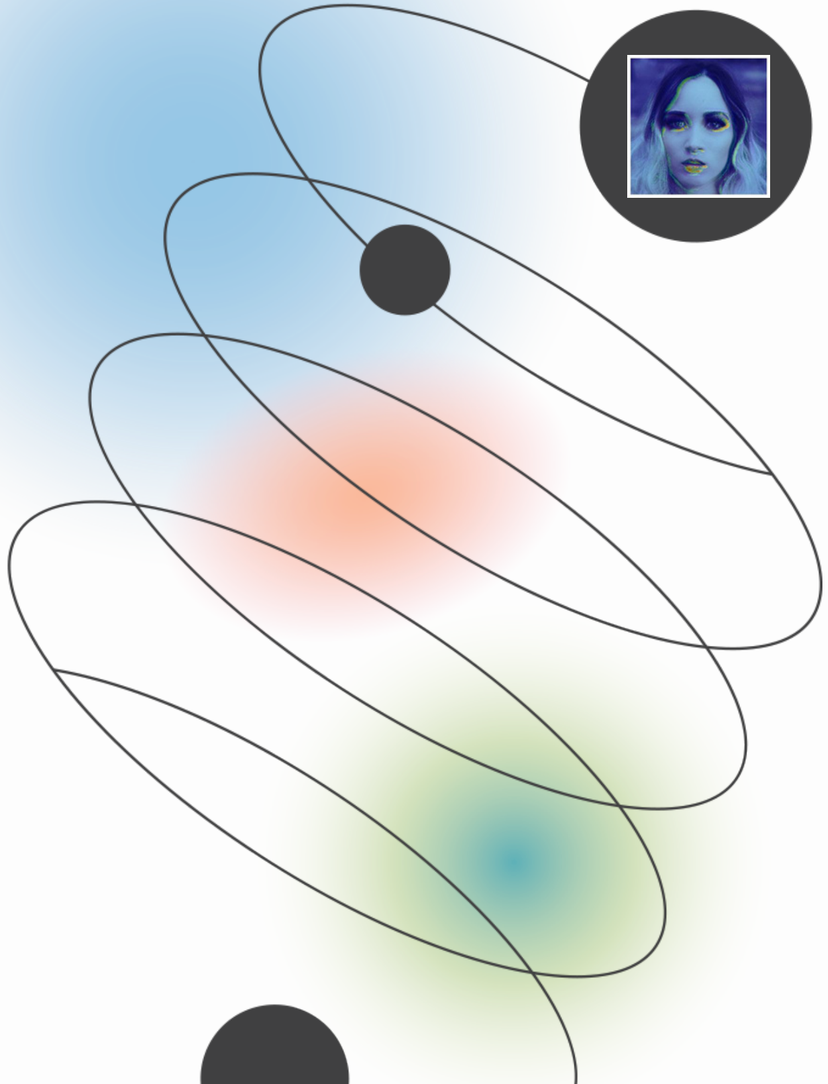
Fake Faces Detection Project



by Radchenko Gleb
DSI-822

Problem statement

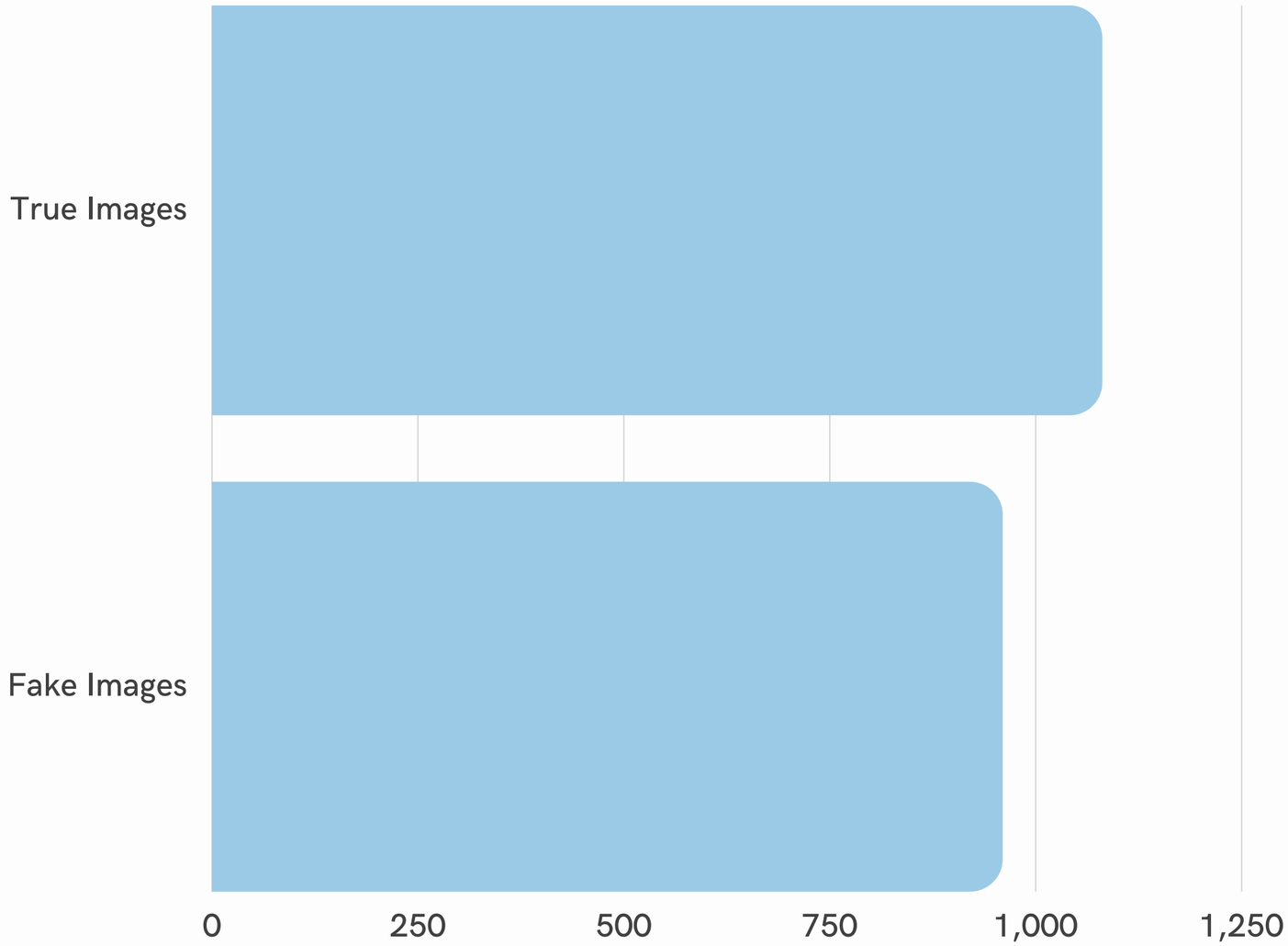
In the past few years, there has been a significant increase in the creation of fake accounts with artificially generated images on the Facebook. As a data scientist working for Meta, I have been tasked to solve this issue by building a classifying model that could detect fake images.

Project success metrics:
accuracy > baseline (53%)

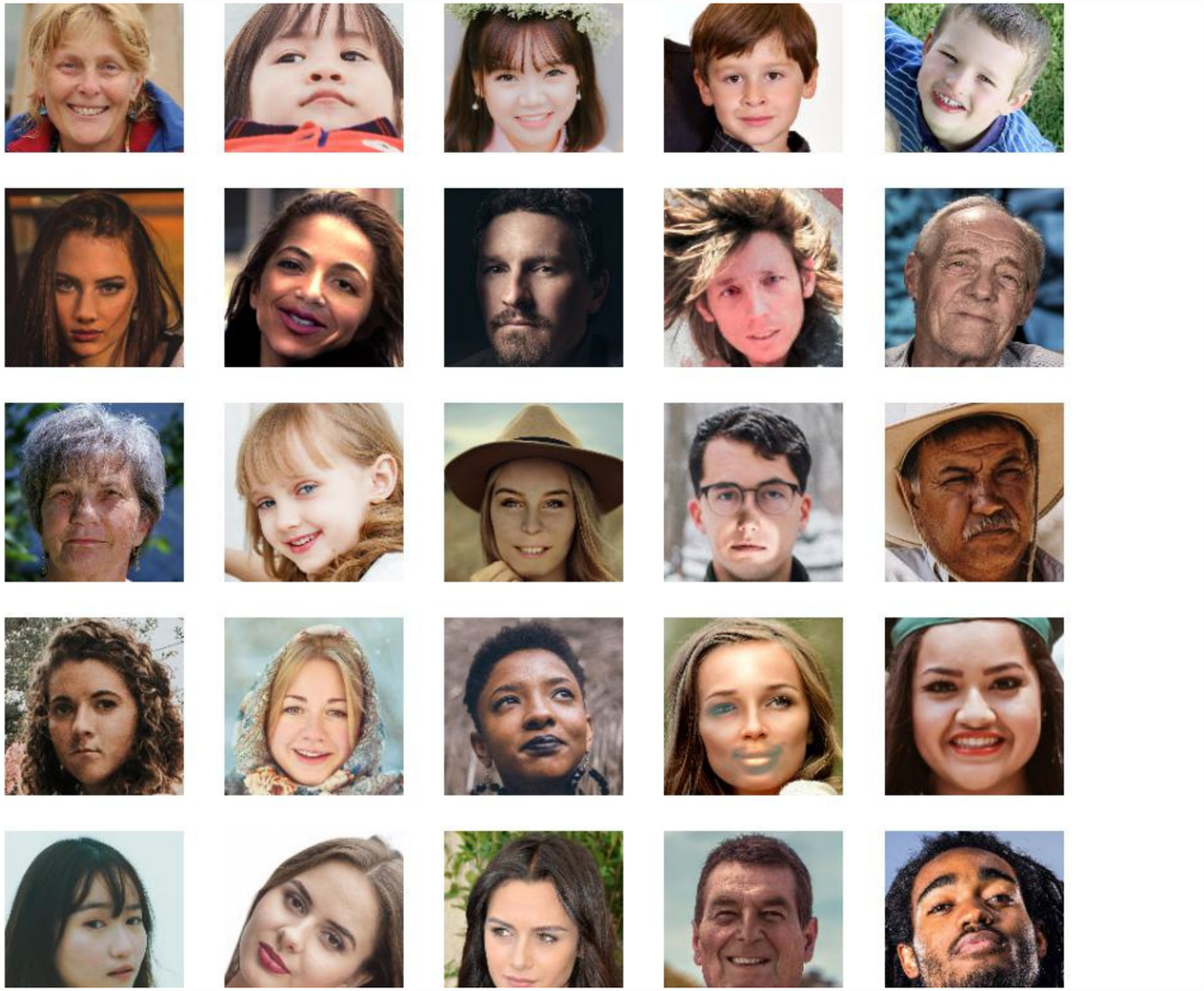
Class proportions

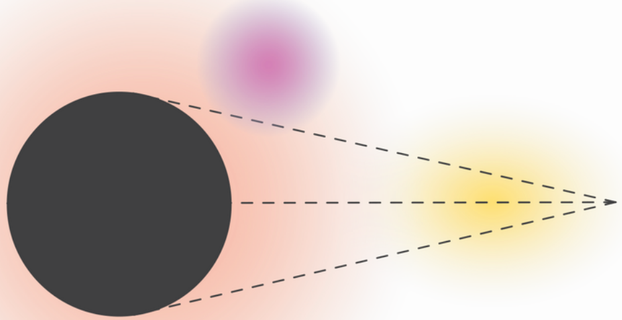
X_train	X_val	X_test
1634	202	205
~80%	~10%	~10%

2041 in total



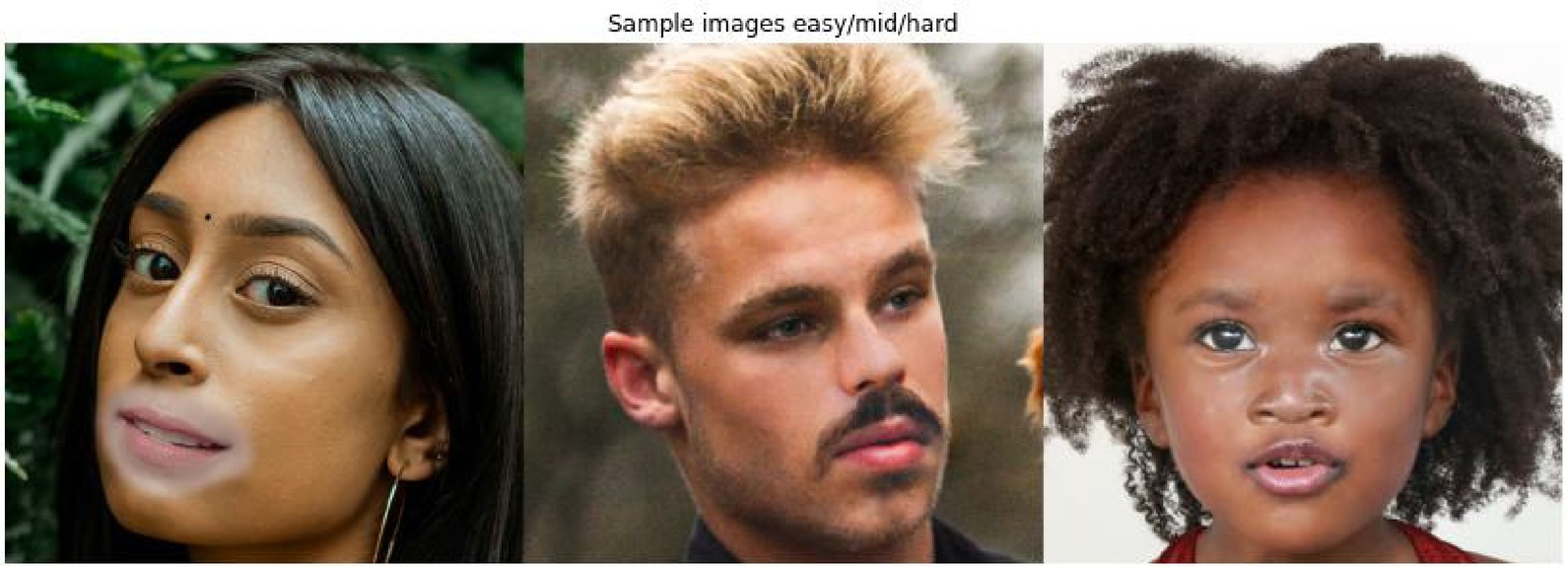
This is how dataset looks like





Fakes and real images together

3 types:



01

Easy to detect

02

Mid tier

03

Hard

Metadata check

AdobeStock_231820750 Properties

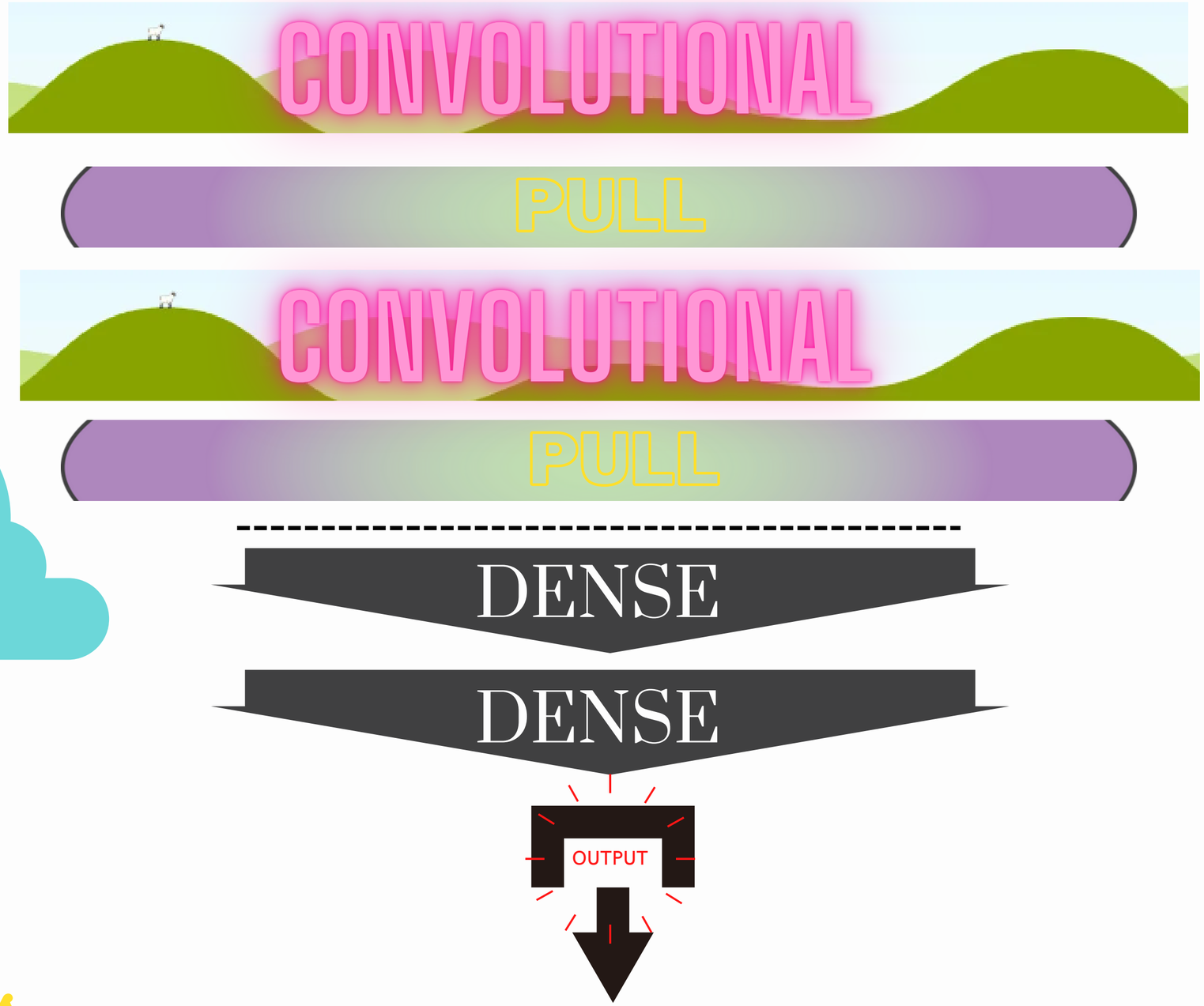
General | Security | Details | Previous Versions

Property	Value
Description	
Title	Mental health image. Various em...
Subject	
Rating	☆ ☆ ☆ ☆ ☆
Tags	mental; health; concept; brain; he...
Comments	
Origin	
Authors	
Date taken	10/19/2018 9:47 PM
Program name	Adobe Photoshop CS6 (Windows)
Date acquired	
Copyright	©tadamichi - stock.adobe.com
Image	
Image ID	
Dimensions	7360 x 4912
Width	7360 pixels
Height	4912 pixels
Horizontal resolution	300 dpi
Vertical resolution	300 dpi

Remove Properties and Personal Information

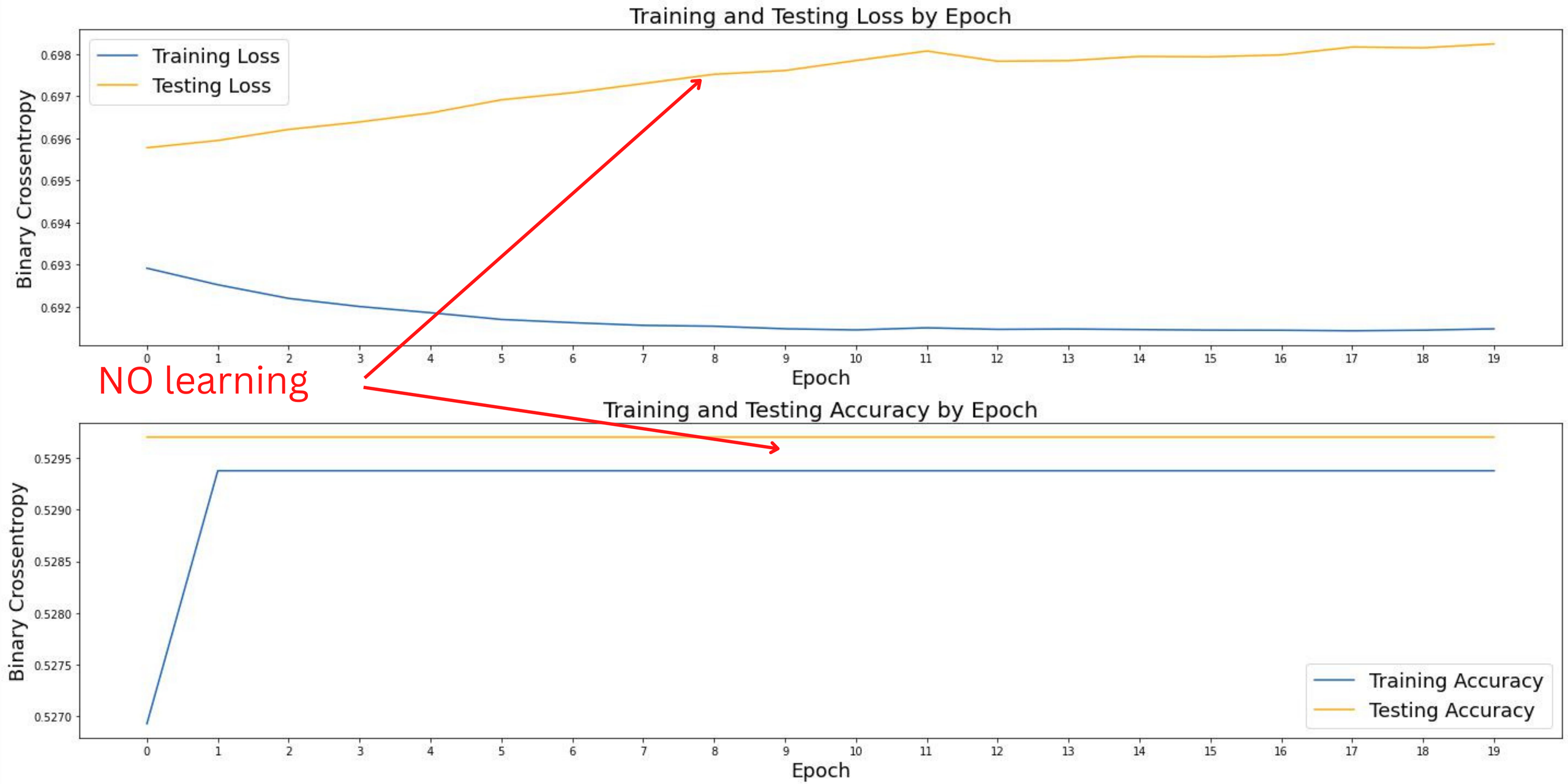
OK | Cancel | Apply

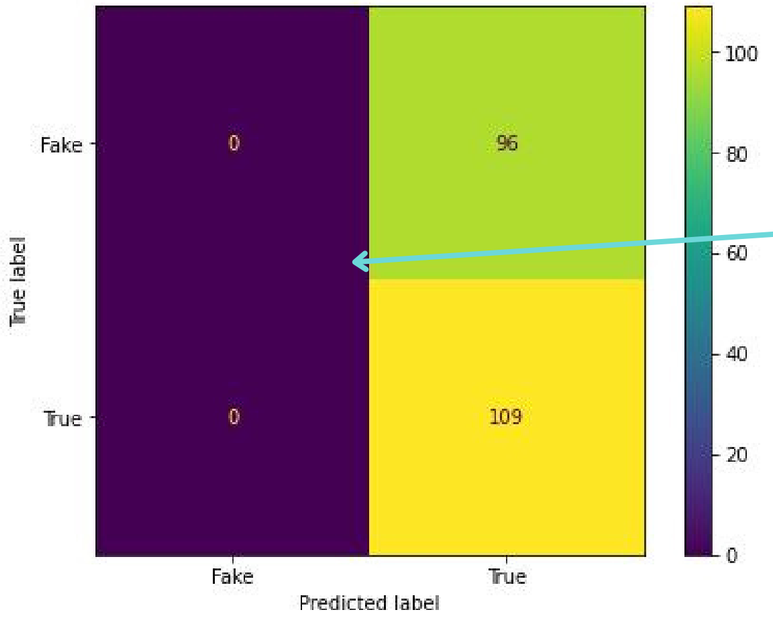
First attempt



Neural networks for beginners

Very simple CNN

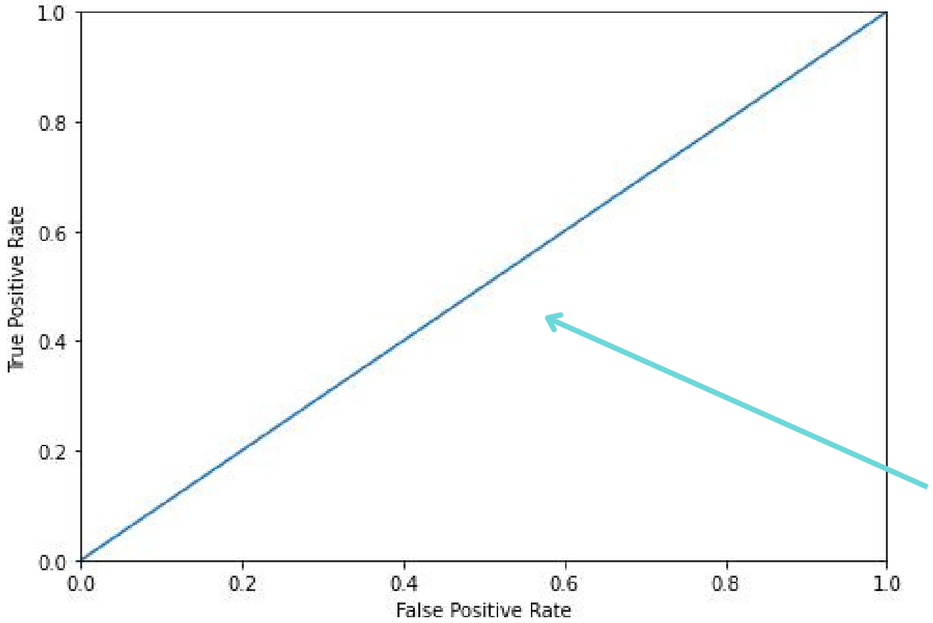




No correctly predicted fakes!

Zero fake images predictions

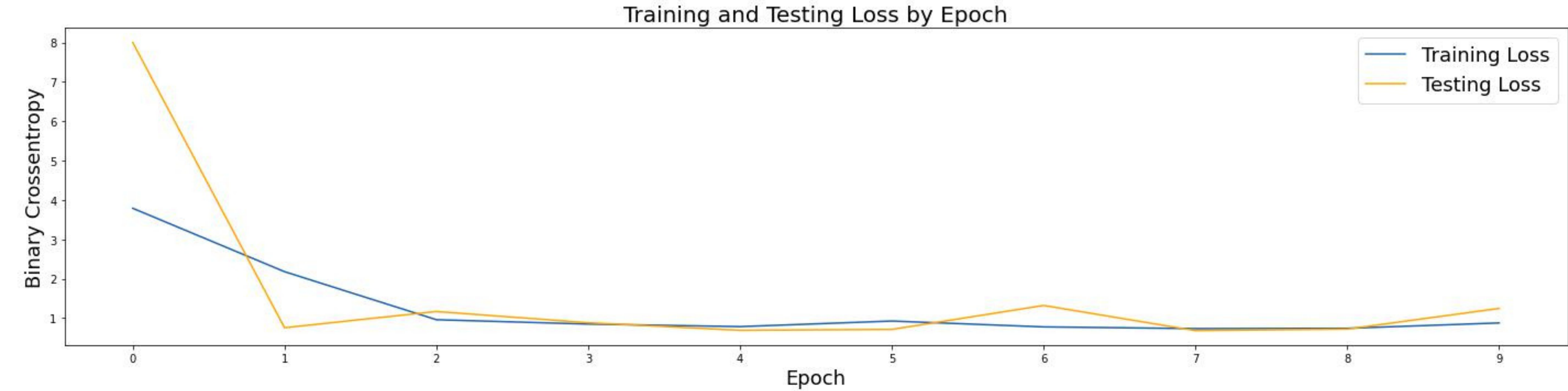
No incorrectly predicted fakes!

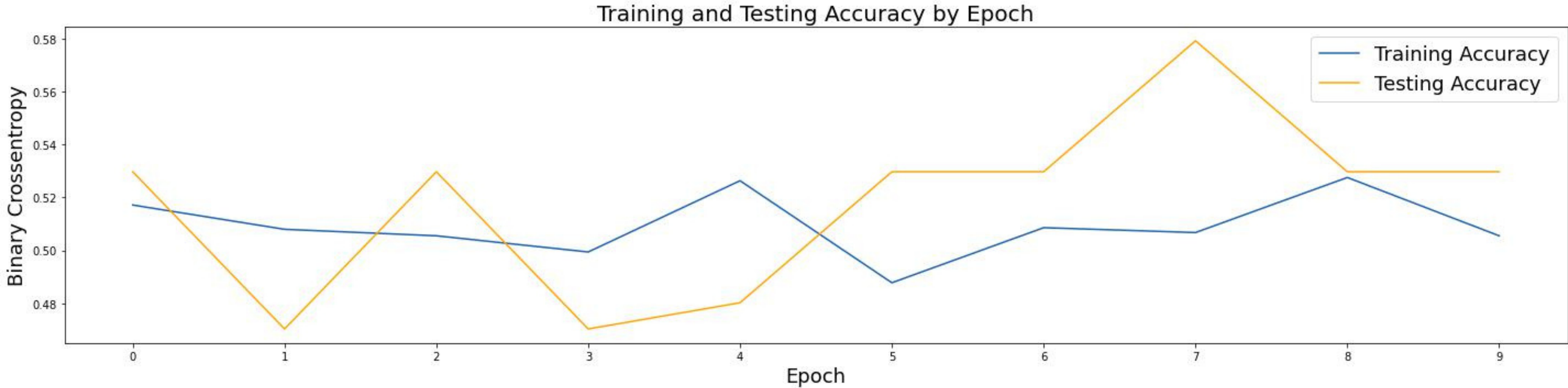


F1 Score : 0.694267515923567
Precision : 0.531707317073170 8
Recall : 1.0
AUC : 0.5
Accuracy: 0.531707317073170 8

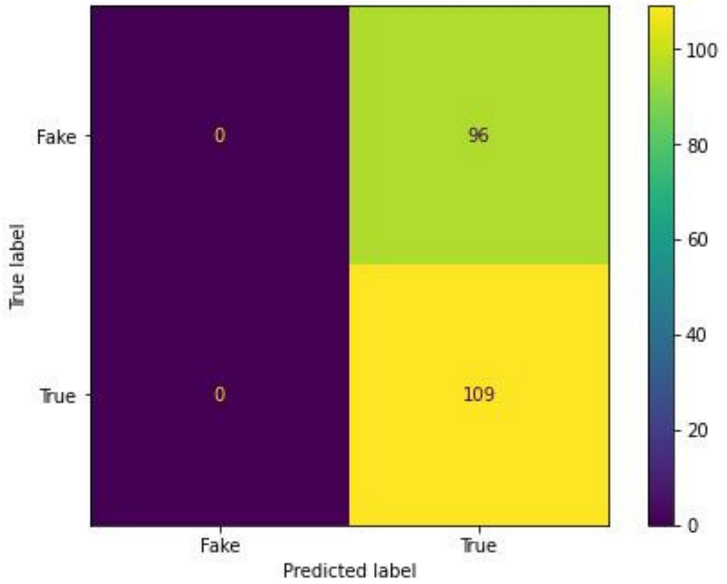
Both classes are identical for the model

Powerful EfficientNetB4



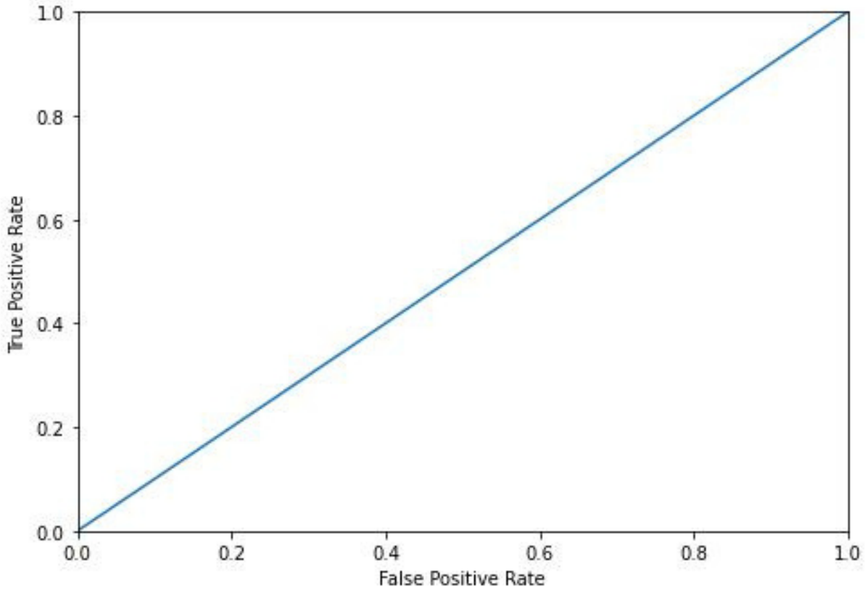


Powerful EfficientNetB4



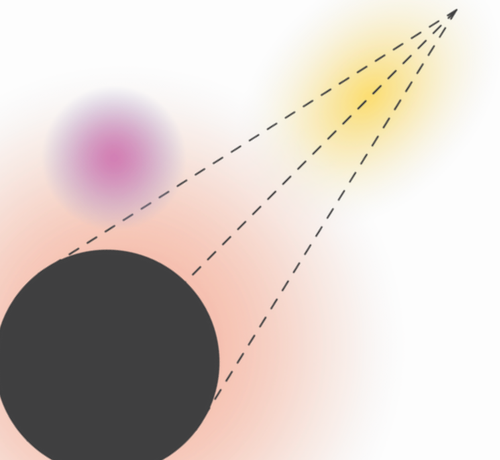
No correctly predicted fakes!

No incorrectly predicted fakes!



F1 Score : 0.694267515923567
Precision : 0.531707317073170 8
Recall : 1.0
AUC : 0.5
Accuracy: 0.531707317073170 8

Custom model



Conv2D(filters=8, kernel_size=(3, 3)

BatchNormalization

MaxPooling2D(pool_size=(2, 2))

Conv2D(filters=8, kernel_size=(5, 5))

BatchNormalization

MaxPooling2D(pool_size=(2, 2))

Conv2D(filters=16, kernel_size=(5, 5))

BatchNormalization

MaxPooling2D(pool_size=(2, 2))

Conv2D(filters=16, kernel_size=(5, 5))

BatchNormalization

MaxPooling2D(pool_size=(4, 4))

Flatten

Dropout(0.5)

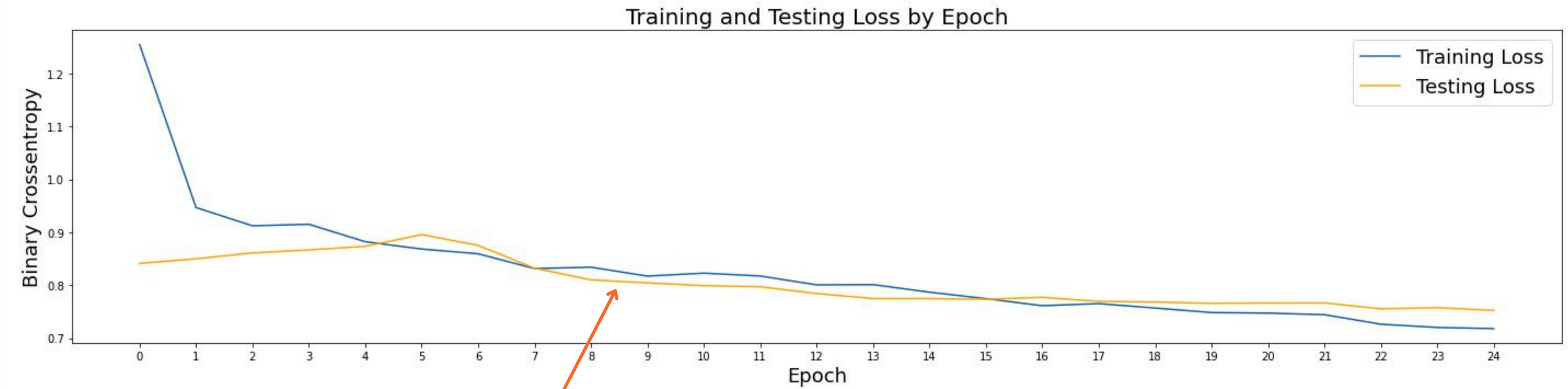
Dense(16,kernel_regularizer =tf.keras.regularizers.l2(l=0.005))

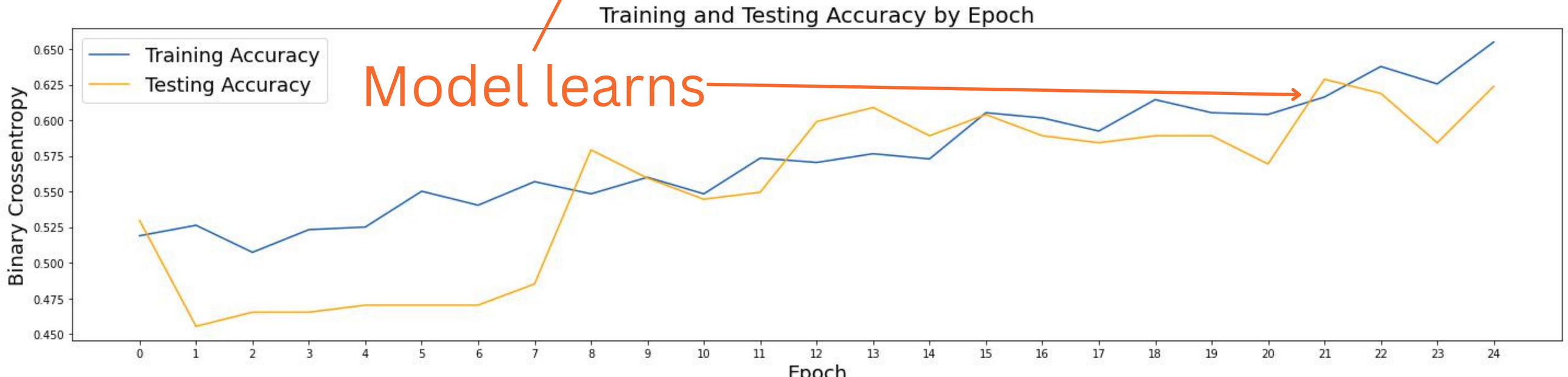
LeakyReLU(alpha=0.1)

Dropout(0.5)

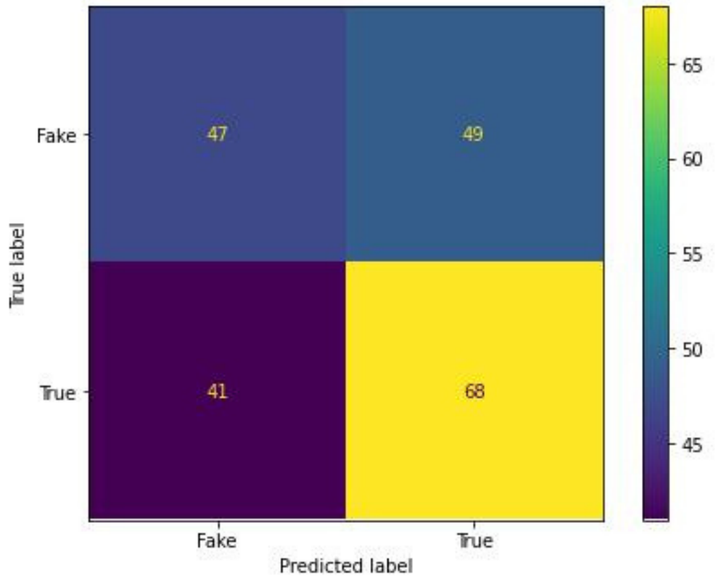
Dense(1, activation = 'sigmoid')

Custom model





Model learns

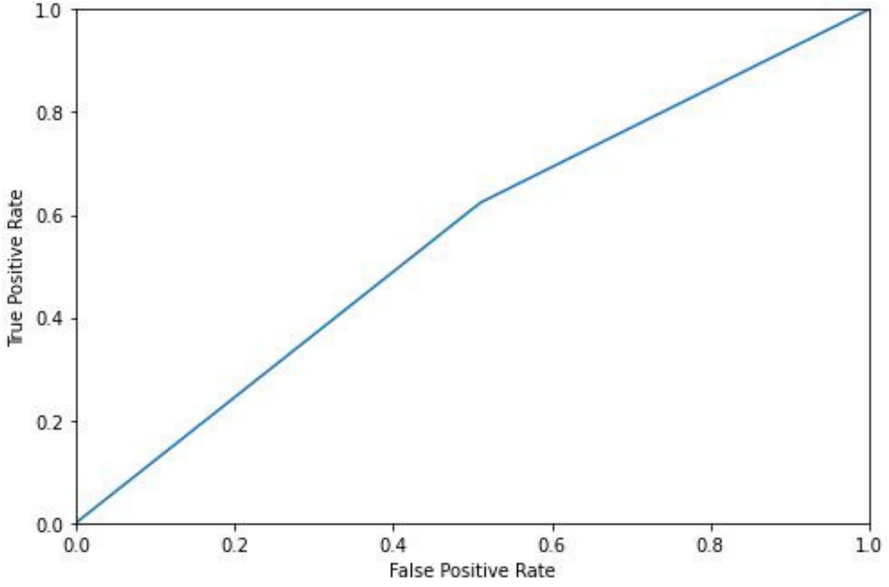


Correctly predicted fakes




Incorrectly predicted fakes



F1 Score : 0.6017699115044248
Precision : 0.5811965811965812
Recall : 0.6238532110091743
AUC : 0.5567182721712538
Accuracy: 0.5609756097560976

Correctly predicted as fake images









Incorrectly predicted as true images





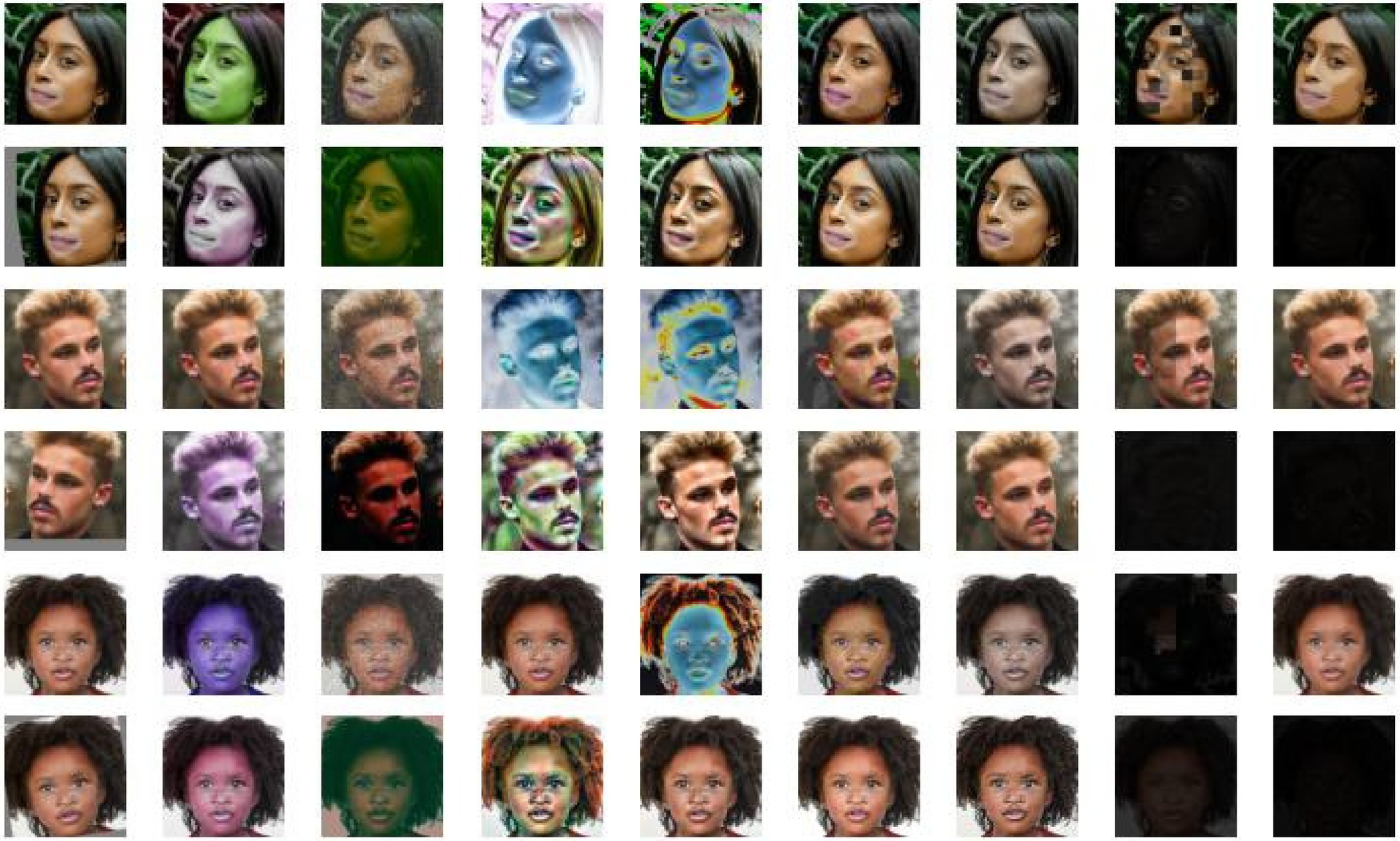




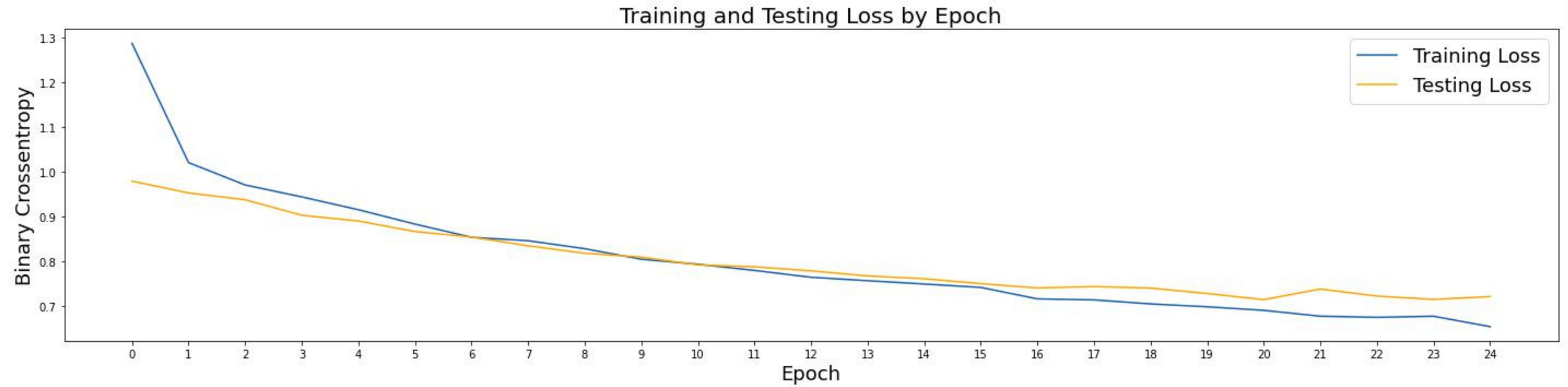
Benefits of augmentations

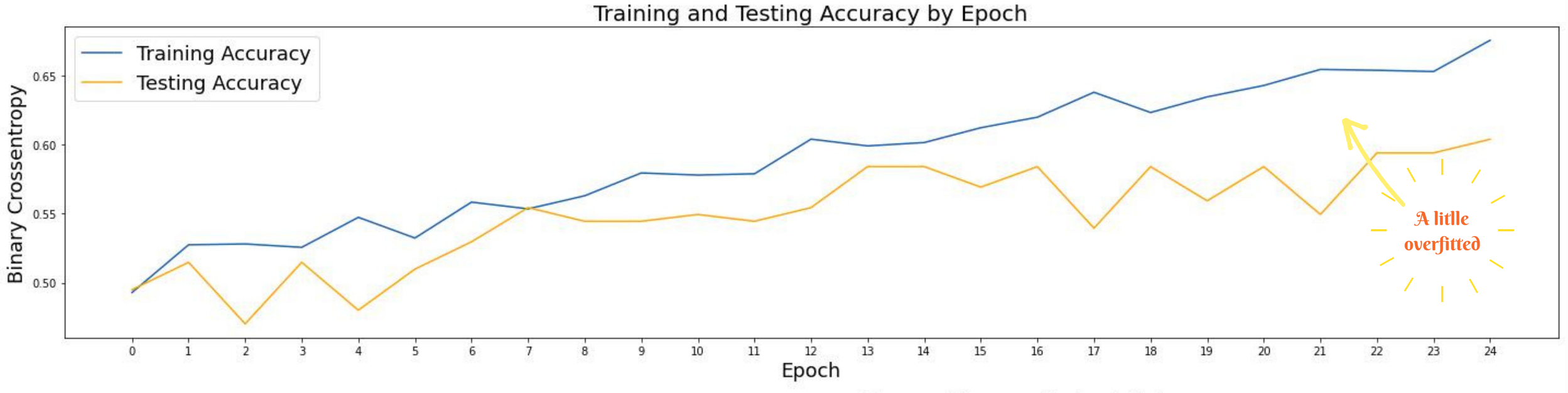
Looks like modern art...

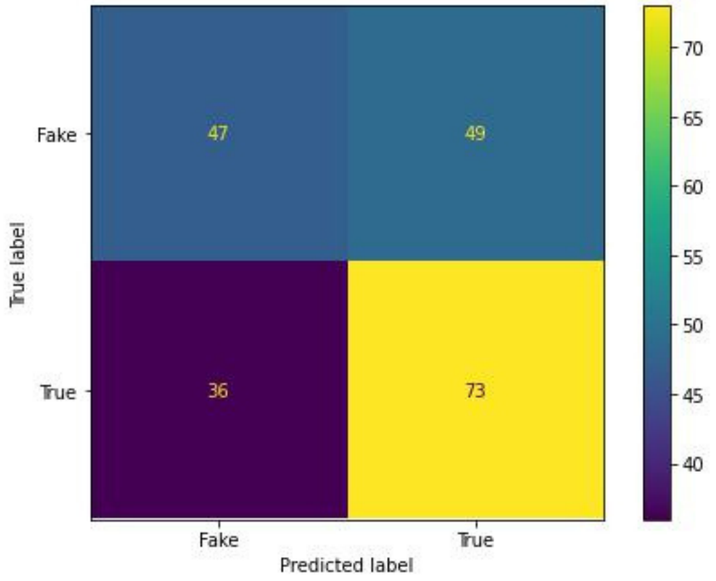
Examples of chosen augmentations



Results after augmentation

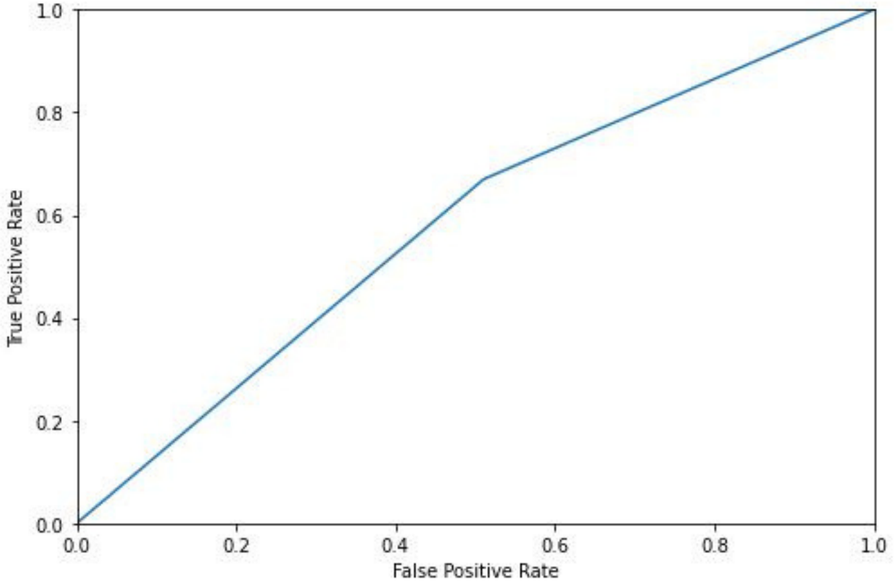




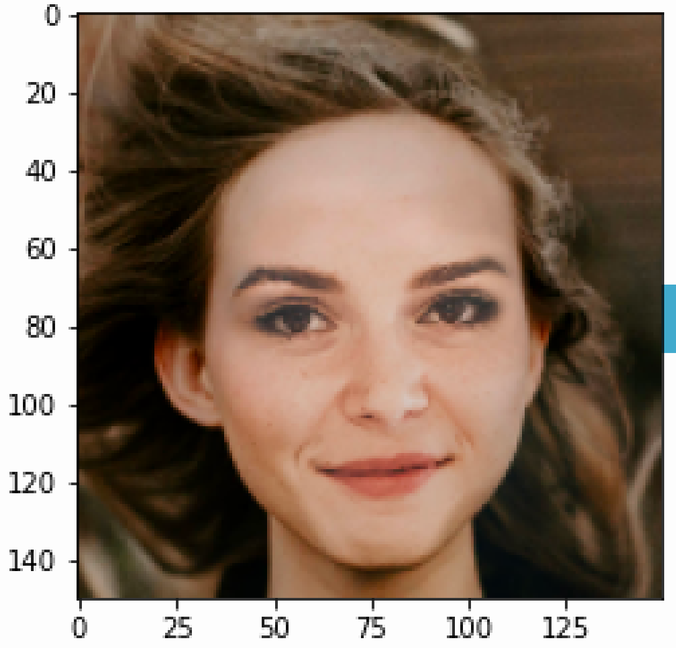


Correctly predicted fakes

Incorrectly predicted fakes

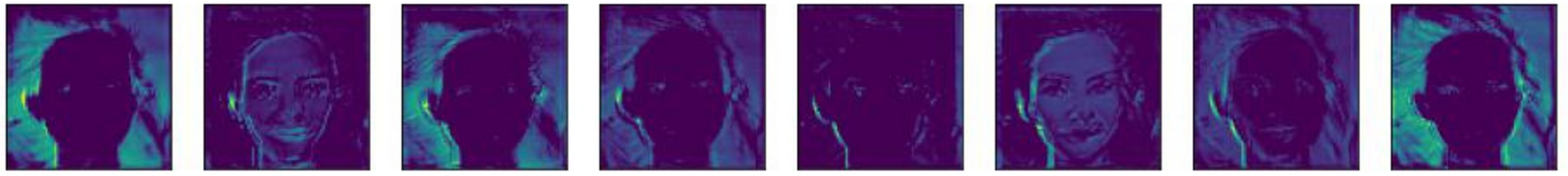


F1 Score : 0.6320346320346321
Precision : 0.5983606557377049
Recall : 0.6697247706422018
AUC : 0.5796540519877675
Accuracy: 0.5853658536585366

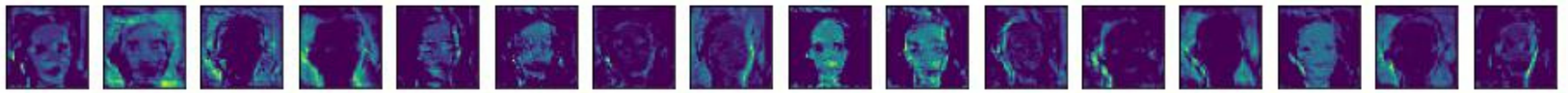


Convolutional layer features

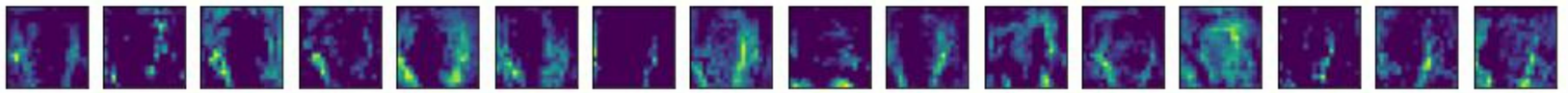
1 Convolutional Layer



2 Convolutional Layer

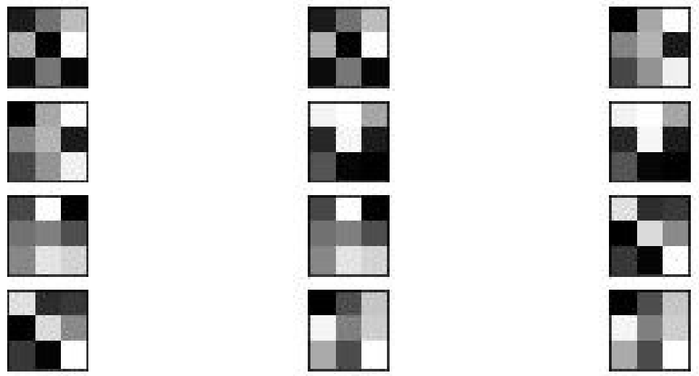


3 Convolutional Layer



4 Convolutional Layer

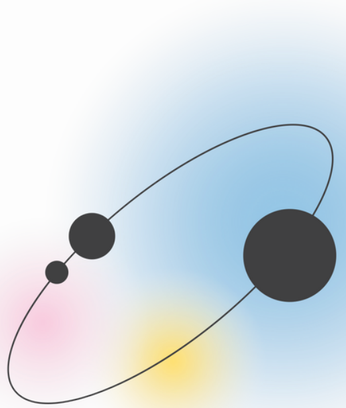
Filters

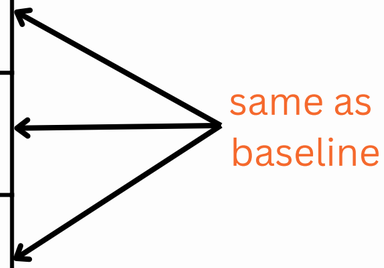


GradCAM

Results

Model name	Accuracy
Simple Convolution Neural Network (baseline)	0.53
VGG Pretrained Convolution Neural Network	0.53
EfficientNetB4 Pretrained Convolution Neural Network	0.53
Custom Convolution Neural Network	0.56
Custom Convolution Neural Network with Augmentations	0.59





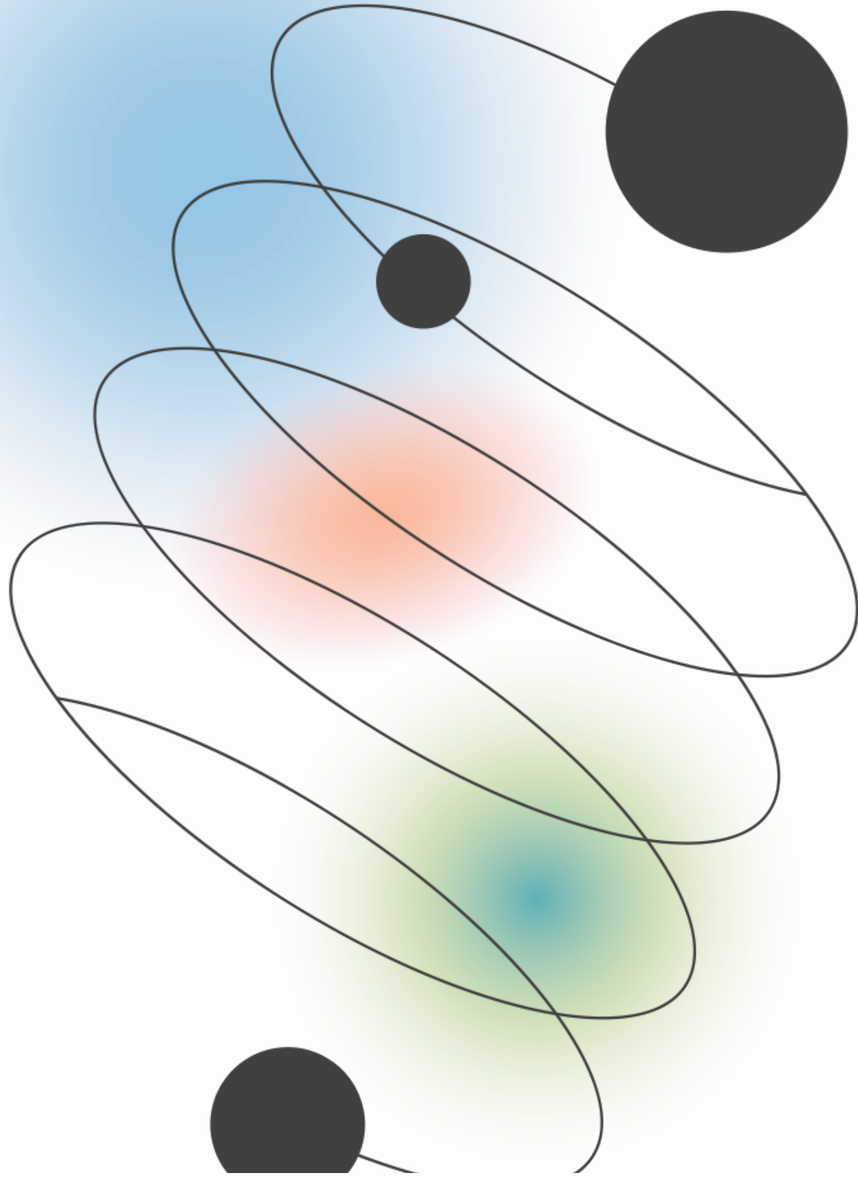
References:

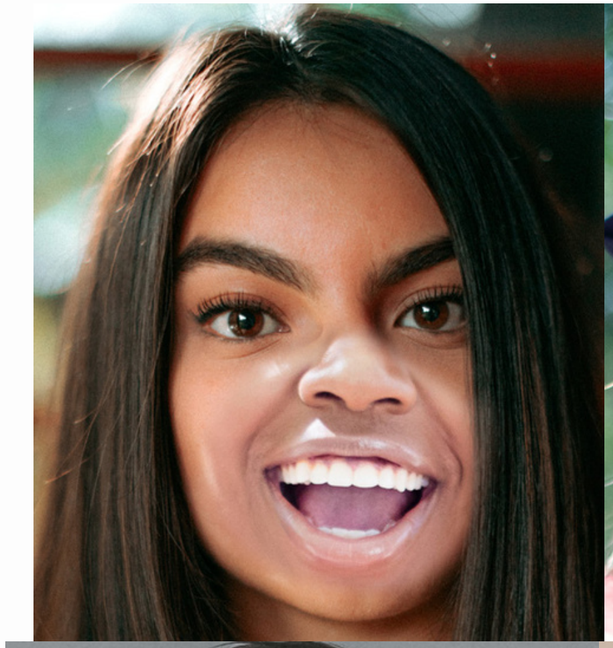
https://github.com/aleju/imgaug

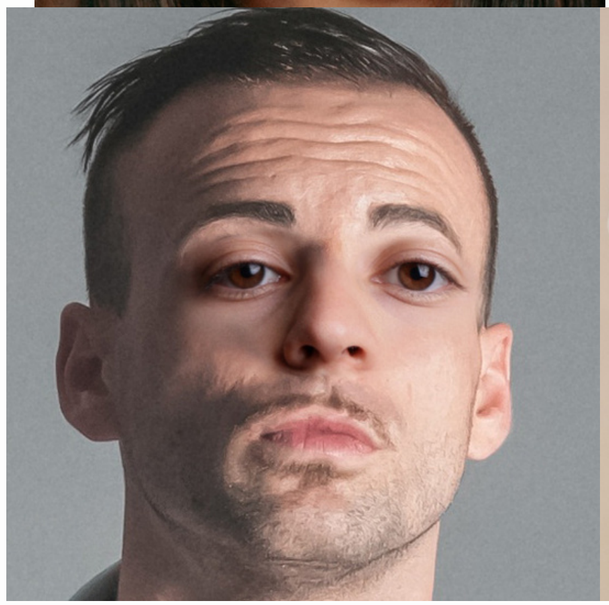
https://github.com/nii-yamagishilab

https://arxiv.org/pdf/2101.03275.pdf

https://machinelearningmastery.com/
how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/








Thank you for your time!

Any questions?