Politechnika
Śląska

Final project report

2024/2025

# Deep learning in theory and aplications

*Detecting AI-Generated Faces.*

Field of study: Informatics

Dominika Wiśniewska

Gliwice, 2024/2025

# Contents

# 1 Problem Definition

Objective: Design and implement a machine learning system capable of predicting whether a presented image of a face is AI-generated.

AI-generated faces have become highly realistic, making it increasingly difficult to distinguish them from real human faces.

Today everyone can use one of many tools to generate an AI image, but there is no requirement for them to be labeled as such.

It is especially a problem when such images are used with malicious intent. This poses significant risks, including identity theft, misinformation, and fraud.

## 1.1 Dataset

Dataset Details:

The dataset contains two folders.

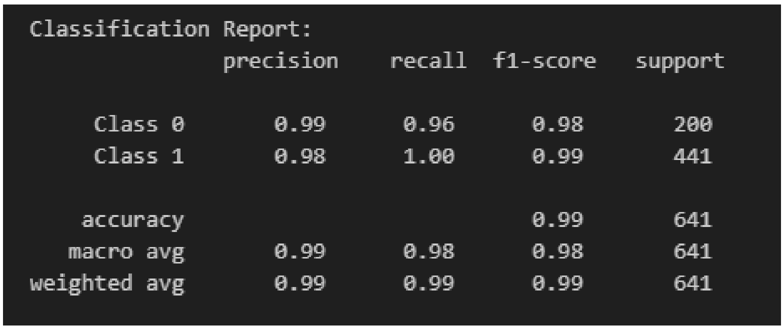
- AI - AI-generated faces
- Real - real faces

## 2 Reasons behind design selections

```
l1 = base_model.output
l2 = MaxPooling2D()(l1)
l3 = Flatten()(l2)
l4 = Dense(128, activation='relu')(l3)
output = Dense(1, activation='sigmoid')(l4)

model = Model(inputs=base_model.input, outputs=output)

model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

- ResNet50 as Base Model: Chosen for its powerful feature extraction capabilities and pretrained weights on ImageNet, reducing the need for extensive training.
- Frozen Layers: Freezing the base model layers ensures that only the custom classification layers are trained, preventing overfitting and speeding up convergence.
- MaxPooling2D: Reduces spatial dimensions while retaining key features.
- Flatten Layer: Converts the feature maps into a one-dimensional vector, allowing for better feature extraction before classification.
- Dense Layers (128 neurons): Helps capture complex relationships before making the final prediction.
- ReLU Activation: Introduces non-linearity in the dense layer, improving learning capacity and model performance.
- Binary Cross-Entropy Loss: Used because our problem is binary classification (real vs. AI-generated faces).
- Adam Optimizer: Selected for its adaptive learning rate, leading to faster and more stable convergence.

I reached the final version of the model by trial and error. I was testing different combinations of layers and activation functions.

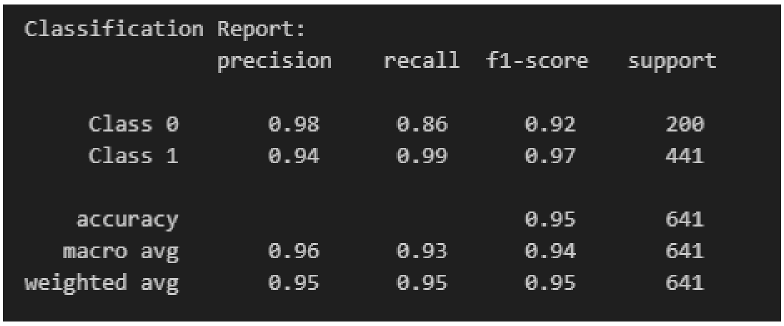


```
Classification Report:
               precision    recall  f1-score   support

     Class 0       0.99      0.96      0.98       200
     Class 1       0.98      1.00      0.99       441

    accuracy                           0.99       641
   macro avg       0.99      0.98      0.98       641
weighted avg       0.99      0.99      0.99       641
```

Figure 1: Classification report of original the model

# 3 Comparing different selections

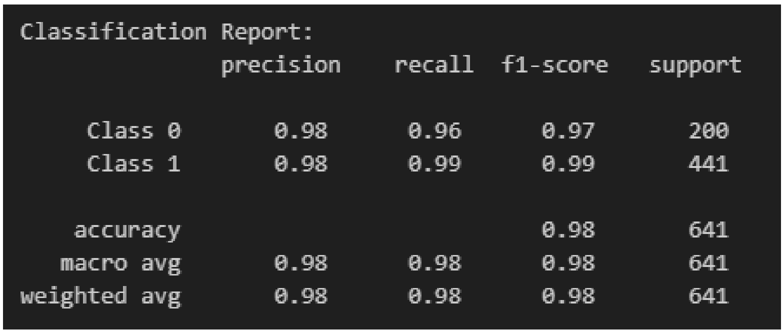
GlobalAveragePooling2D instead of MaxPooling2D



```
Classification Report:
               precision    recall  f1-score   support

     Class 0       0.98      0.86      0.92       200
     Class 1       0.94      0.99      0.97       441

    accuracy                           0.95       641
   macro avg       0.96      0.93      0.94       641
weighted avg       0.95      0.95      0.95       641
```

Figure 2: Classification report of the model with GlobalAveragePooling2D

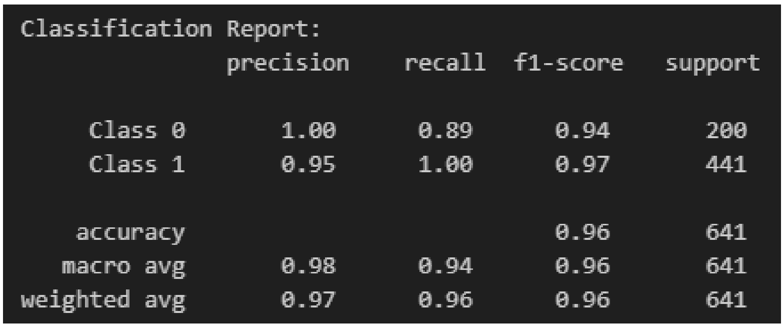
LeakyRelu instead of Relu



```
Classification Report:
               precision    recall  f1-score   support

     Class 0       0.98      0.96      0.97       200
     Class 1       0.98      0.99      0.99       441

    accuracy                           0.98       641
   macro avg       0.98      0.98      0.98       641
weighted avg       0.98      0.98      0.98       641
```

Figure 3: Classification report of the model with LeakyRelu

Swish instead of Relu



```
Classification Report:
              precision    recall  f1-score   support

     Class 0       1.00      0.89      0.94       200
     Class 1       0.95      1.00      0.97       441

    accuracy                           0.96       641
   macro avg       0.98      0.94      0.96       641
weighted avg       0.97      0.96      0.96       641
```

Figure 4: Classification report of the model with swish