Computer Vision Project

Layer-wise Depth Integration in RGBD Deepfake Detection

Daidone Giuseppe 2122594 – a.y. 2023/2024

Introduction

During the last years, deepfake contents have become more sofisticated and spread over social medias, causing a large impact on social and political choises. Building a strong architecture which is capable of detecting deepfakes is a fundamental task to prevent the spreading of fake content.

Recent studies[1][2] have shown that building a deepfake detector using RGB and Depth component is more robust and accurate.

Objective of the study

Evaluate the impact of Depth integration for a RGBD Deepfake Detector in three different scenarios:

1. Input layer depth
2. Middle layer depth
3. Output layer depth

Final goal: understand which is the best model, comparing them with evaluation metrics (accuracy, precision, recall, f1-score, ROC curve).

Dataset

FaceForensics++ Dataset[3]: collection of 1000 youtube videos (original and manipulated versions).

Dataset organization:

- Original sequences: original videos (not manipulated)
- Manipulated sequences: manipulated videos with different face forgeries techniques (Deepfake, Face2Face, FaceShifter, FaceSwap, NeutralTextures)

Preprocessing

To prepare the training/validation/test data, it was performed a preprocessing (offline) operation.

The idea is to save frames from dataset videos as RGB images, detect and extract human face and perform monocular depth estimation to estimate the Depth.

Note: for the aim of this study, it was selected the Deepfake face forgery only.

Frame Extractor

The frame extractor is the component that extract the frames from the videos.

In this implementation, one frame is saved every 3s in order to deal with a medium number of images during the training process (~10k RGB).

Data Augmentation

In order to augment the data, different transformations are performed to the frames (with certain probability):

- Vertical flip [30%]
- Random rotation [20%]
- Gaussian noise [10%]
- Salt-and-Pepper noise [10%]

MediaPipe

MediaPipe is a set of libraries and tools developed by Google which provides solutions to many computer vision tasks.

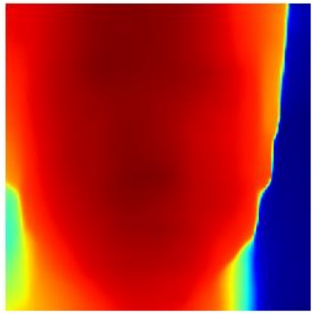
In this study, MediaPipe Face Detector[4] is used to detect and extract face from the video frames in a 224x224x3 image.

MiDaS

MiDaSV3[5] is a monocular depth estimator and the model is available in different versions of accuracy and computational time:

- MiDaS Large
- MiDaS Hybrid (chosen model)
- MiDaS Small

RGB
224x224x3

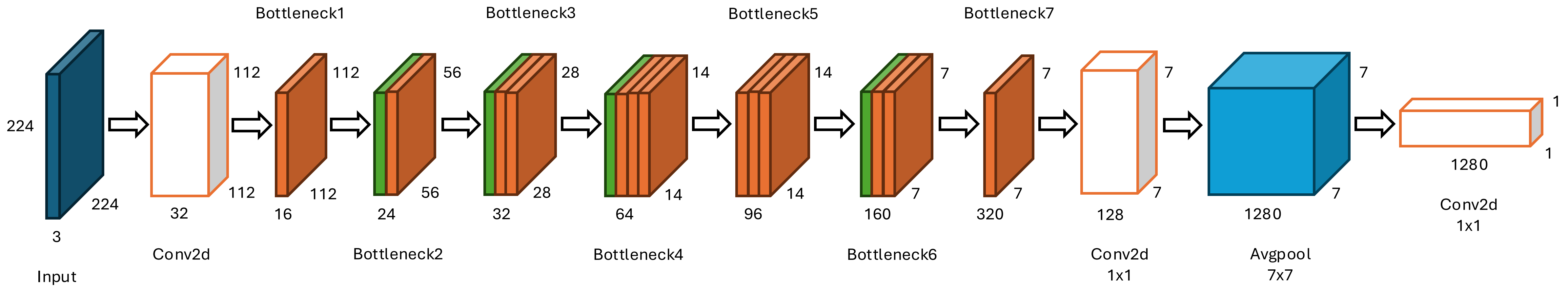


D
224x224x1

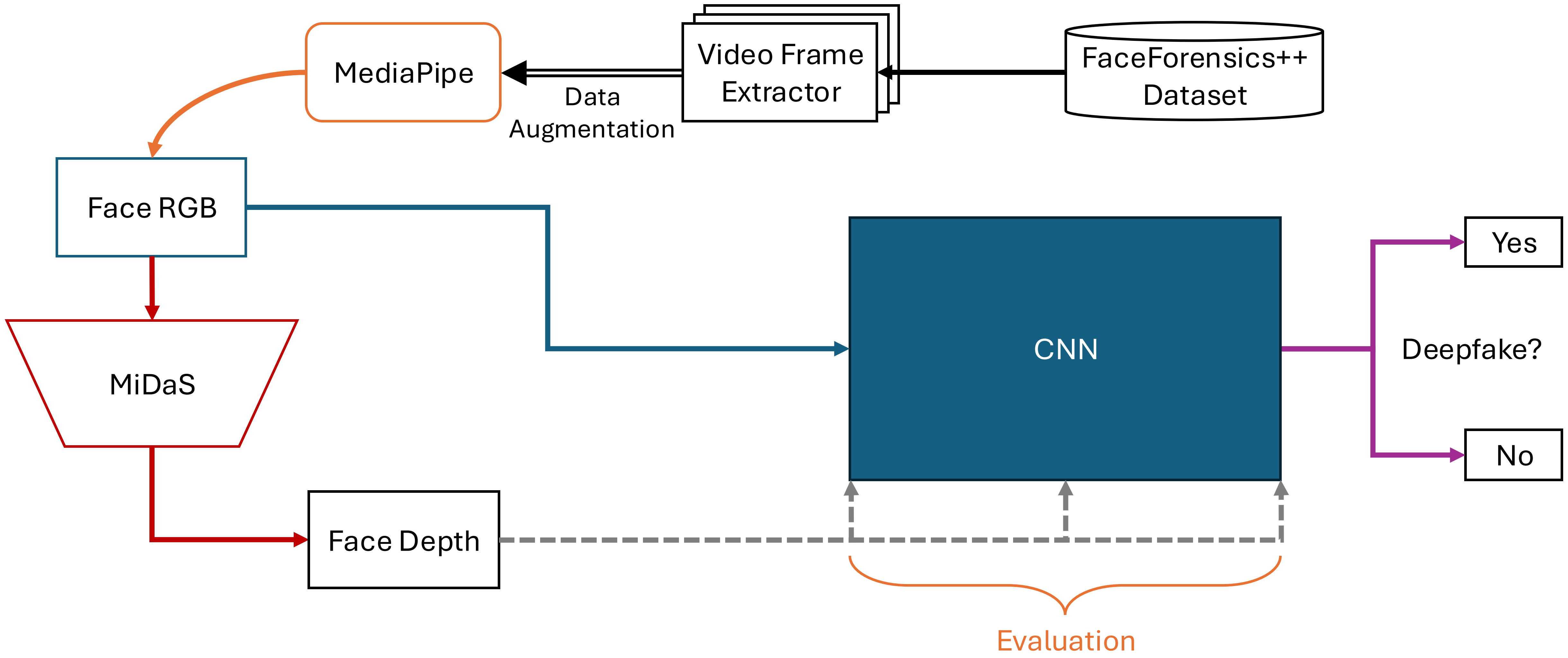
MiDaS is responsible for computing the Depth (normalized 0-225), starting from a RGB image.

CNN

MobileNetV2[6] is a pre-trained CNN (on ImageNet Dataset) developed by Google. The CNN is adapted and modified in order to accommodate different scenarios for the purpose of this study.

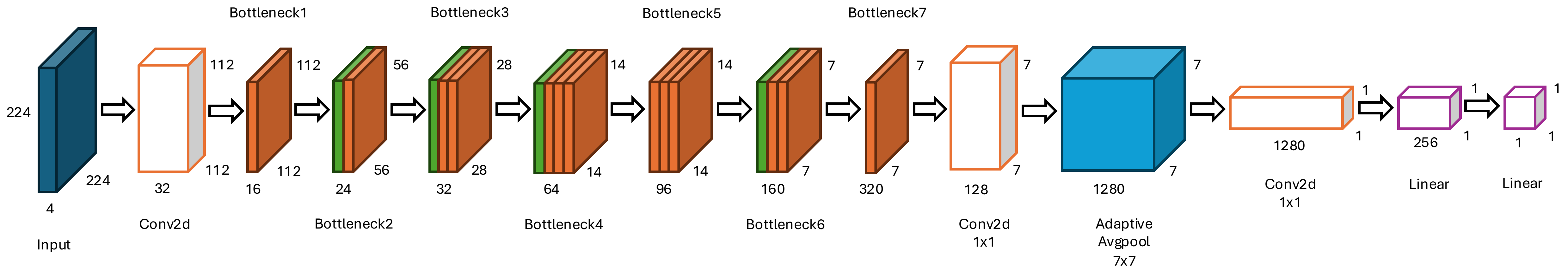


Architecture pipeline



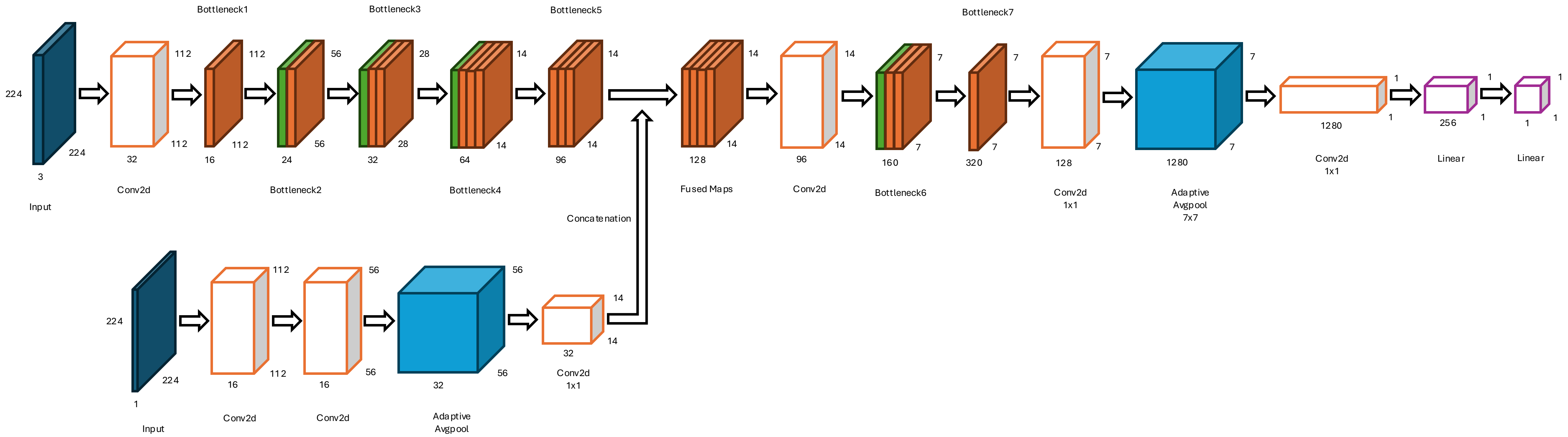
CNN with RGBD as input layer

- The input layer size has been modified to 224x224x4 in order to fit RGBD image
- Adaptive average pooling is explicitly added to ensure that the output is always reduced to a fixed size of 1x1 per channel
- After the convolutional 1x1 layer, the output is a 1280-dimensional feature vector that needs to be reduced to 1x1x1 output for binary classification, so two fully connected layers are added which linearly reduce the output size
- In the last layer, Sigmoid activation function is used for binary classification



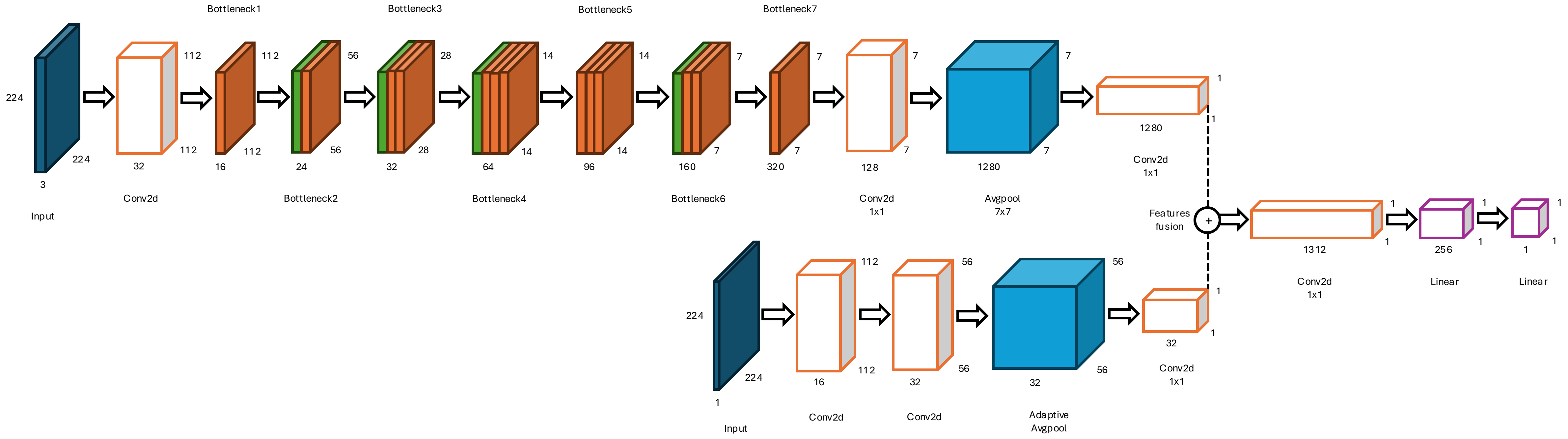
CNN with RGB as input layer, D as mid layer

- RGB image start following the standard MobileNetV2 architecture until bottleneck5
- Depth map goes into a separate CNN where features map are extracted with size of 32x14x14 (batch normalization and ReLU activation function are applied)
- Depth features are concatenated with in the middle of the MobileNet pipeline obtaining 128x14x14 fused maps
- After convolution to format again in 96x14x14, the networks proceeds as the previous architecture
- In the last layer, Sigmoid activation function is used for binary classification



CNN with RGB as input layer, D as output layer

- RGB image follows the standard MobileNetV2 architecture
- Depth map goes into a separate CNN where features map are extracted into a 32-dimensional vector (batch normalization and ReLU activation function are applied)
- At the output of the networks, the features are fused into a 1312-dimensional vector, then two fully connected layers are added which linearly reduce the output size
- In the last layer, Sigmoid activation function is used for binary classification

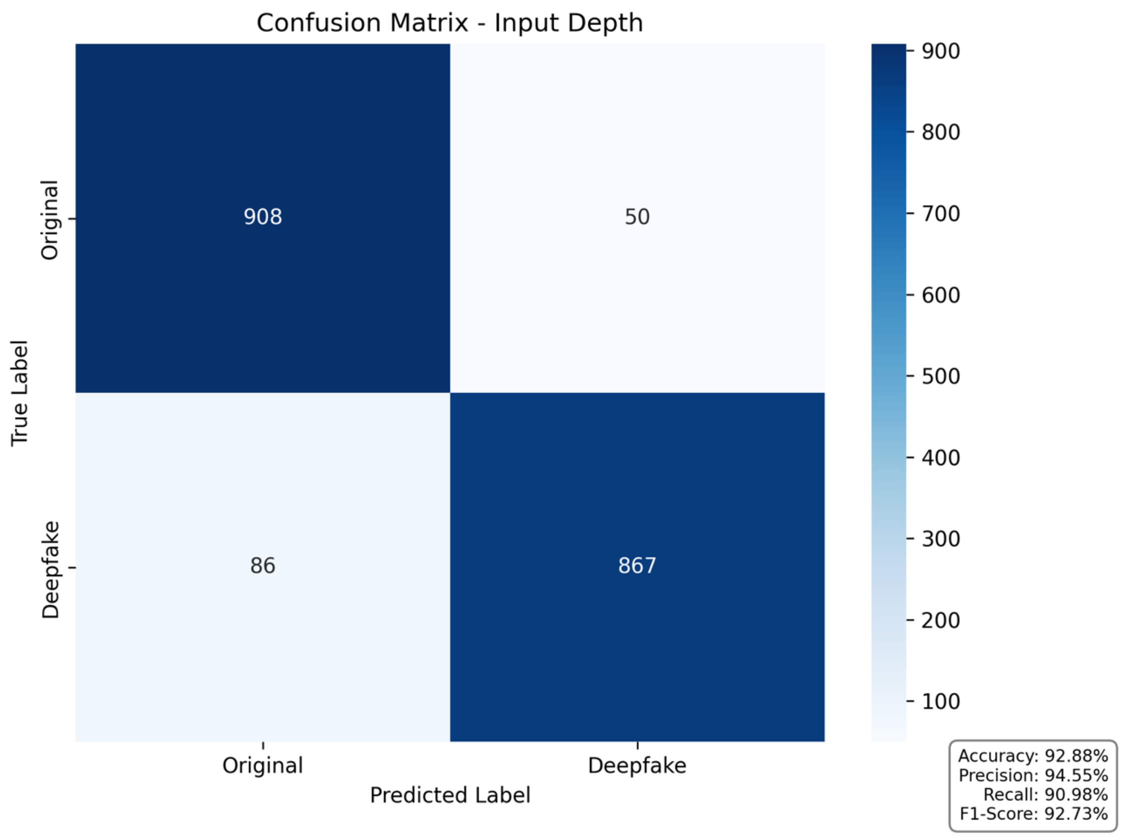


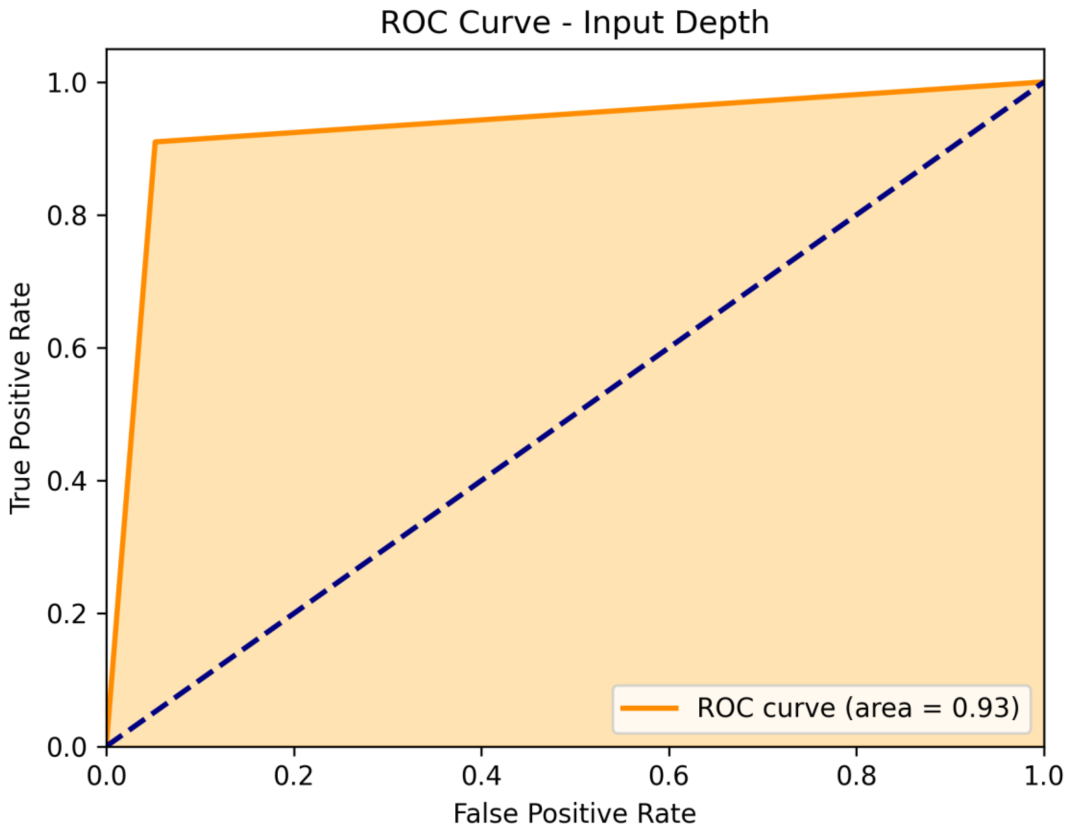
Common implementation details

Common implementation details:

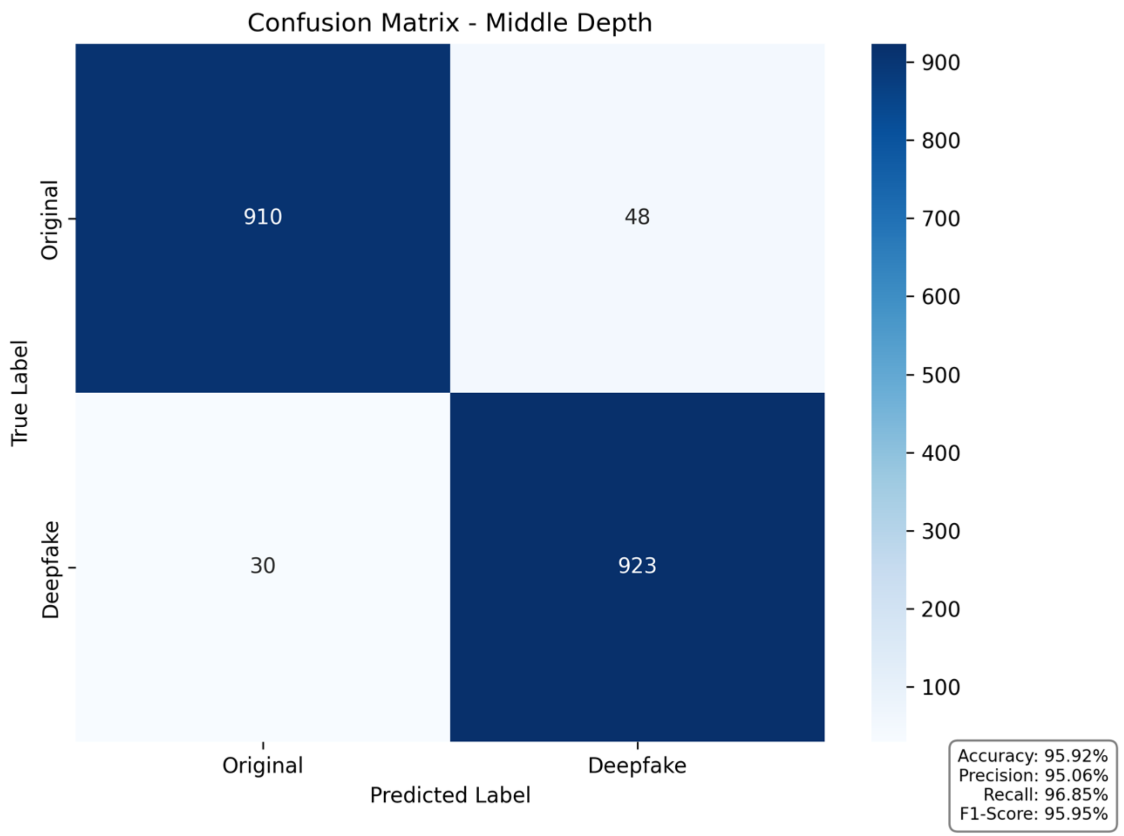
- Loss function: Binary Crossentropy (BCE) Loss
- Optimizer: ADAMAX
- Epochs: 15
- Training/Validation/Testing dataset: 80/10/10

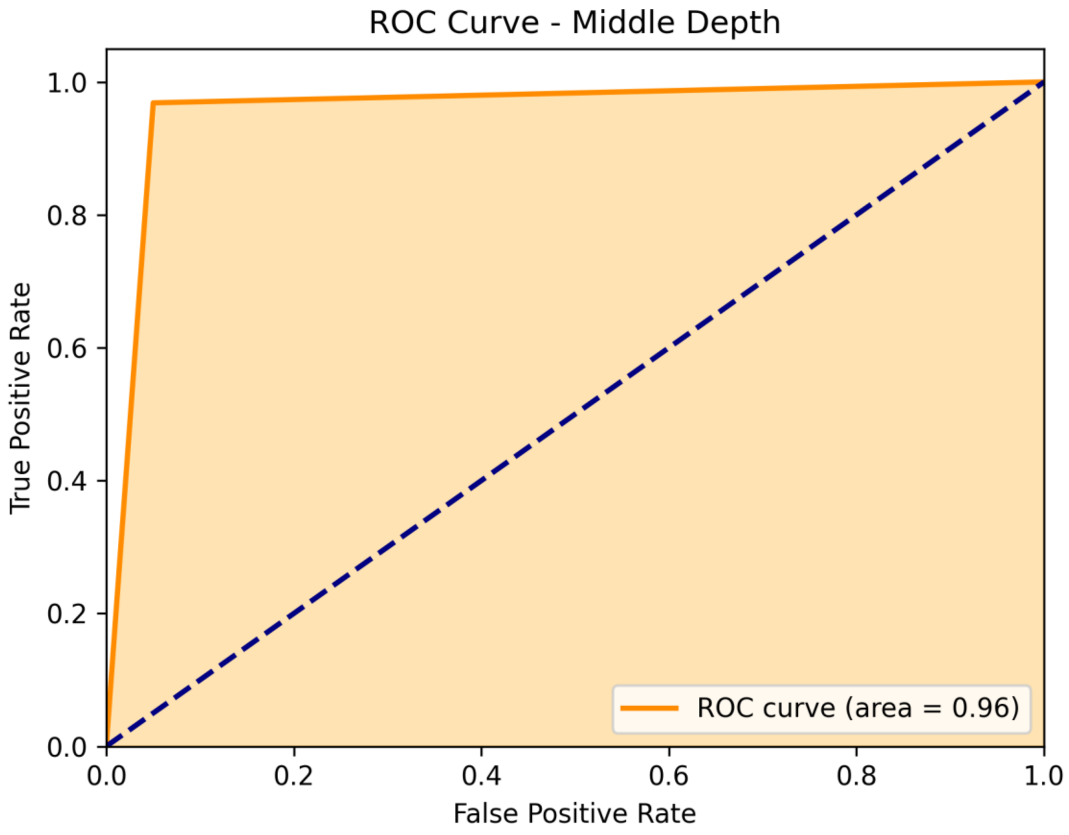
Evaluation: input layer depth



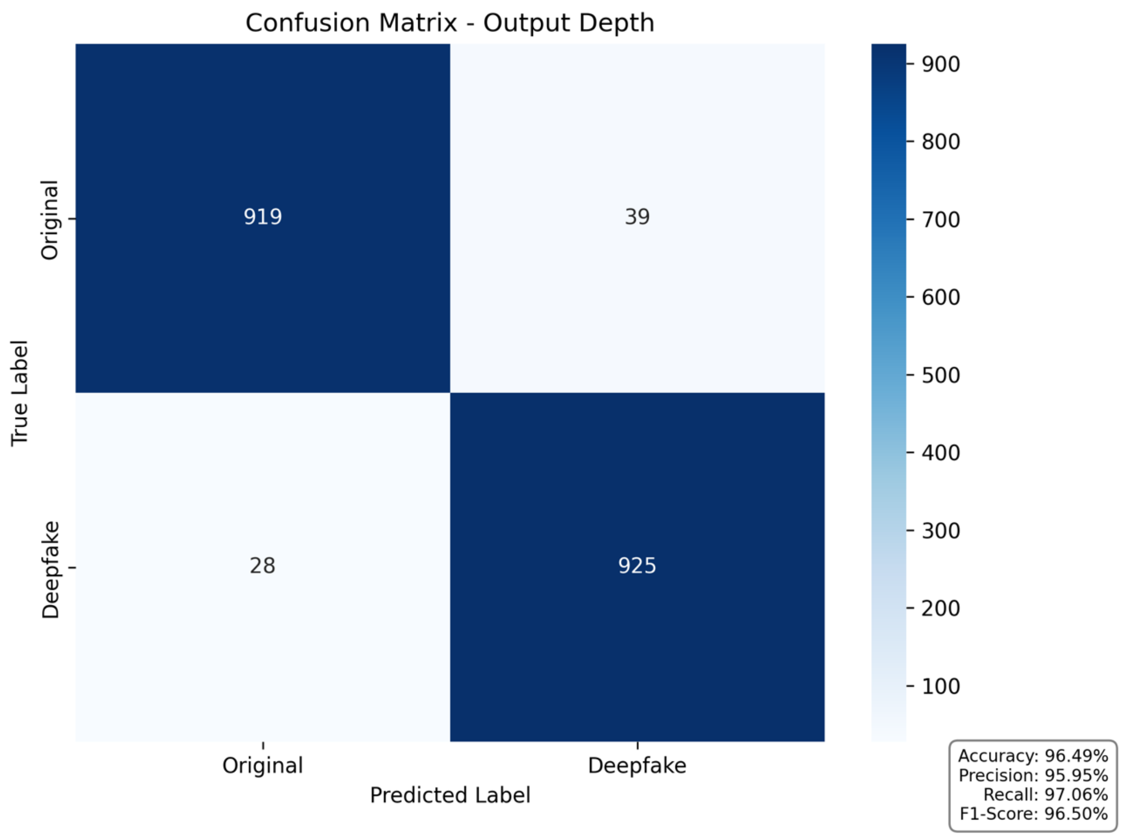


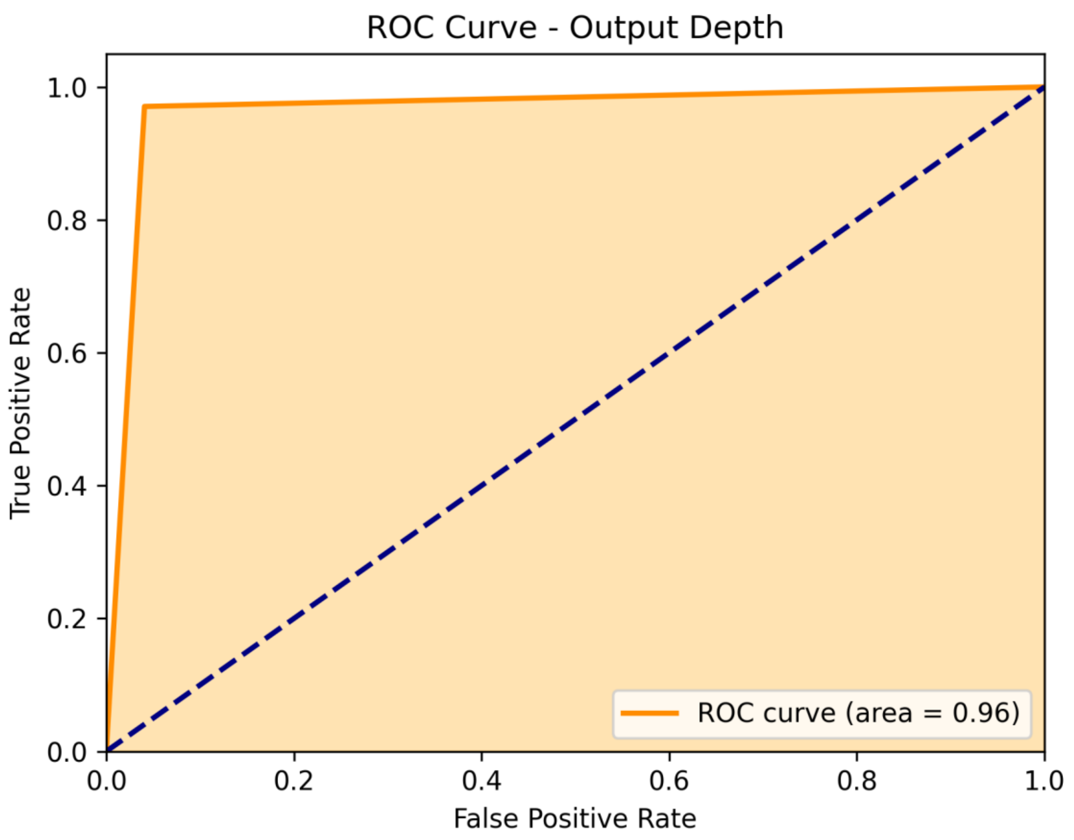
Evaluation: mid layer depth





Evaluation: output layer depth





Comparison between models

	Input D	Mid D	Output D
Accuracy	92.88%	95.92%	**96.49%**
Precision	94.55%	95.06%	**95.95%**
Recall	90.98%	96.85%	**97.06%**
F1-Score	92.73%	95.95%	**96.50%**

*Bold represents the best metric.

Conclusions

Performance metrics show that processing and inserting Depth information in the final layer of the network is the best way to design a more accurate Deepfake Detector.

Bibliography

[1] Maiano, L., Papa, L., Vocaj, K., Amerini, I., 2022. DepthFake: a depth-based strategy for detecting Deepfake videos.
[2] Leporoni, G., Maiano, L., Papa, L., Amerini, I., 2024. A Guided-Based Approach for Deepfake Detection: RGB-Depth Integration via Features Fusion.
[3] https://github.com/ondyari/FaceForensics
[4] https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector
[5] https://github.com/isl-org/MiDaS
[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks.