Project Report - CSPC54

Block Diagrams and Algorithms followed for

Deepfake Face Identification - Comparative Analysis of CNN, Masked CNN and Deformable CNN

Dhruvi Sachin Shah (106122038)

Sanjana Gummuluru (106122106)

Introduction

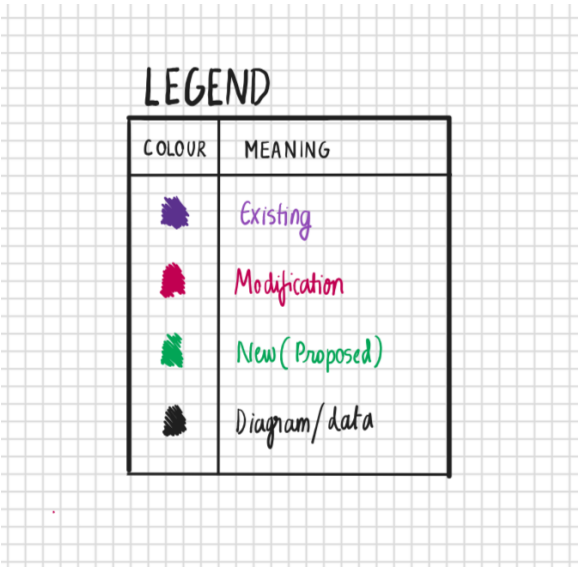
DeepFake, which is a portmanteau of the terms ‘deep learning’ and ‘fake’, is a new vein of AI generated fake videos synthesized using generative ML models. They can achieve high degrees of realism and have thus been used in malignant ways, manipulating people into believing something is real when it is not.

The proposed model for DeepFake Face Detection (Alben Richards MJ et al) uses a CNN-based approach to detect manipulated media, given that CNNs are particularly well suited for image and video analysis tasks.

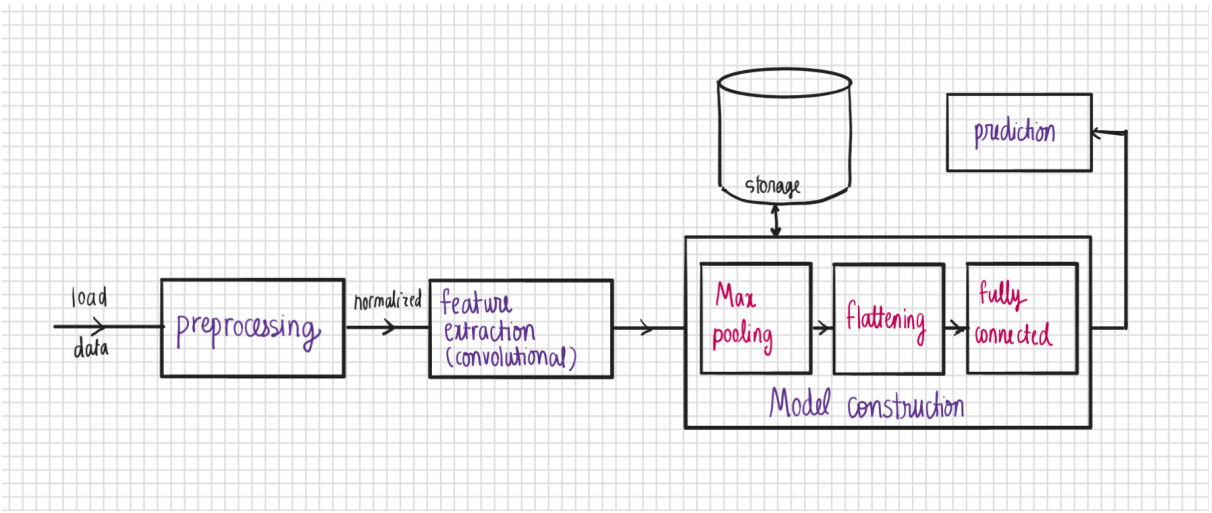
A Masked CNN model would, due to masking, allow the model to learn which aeas of the image are relevant for DeepFake detection, such as unnatural facial features or artifacts. This can result in faster training and inference, as well as reduced overfitting, since irrelevant data is excluded.

A Deformable CNN adds more flexibility in the way convolutional filters interact with the input data. It allows the network to adapt its receptive field and handle geometric variations in input images by learning to deform the convolutional grid. Thus, it would ideally perform better at DeepFake detection, where small facial manipulations are crucial for correct classification.

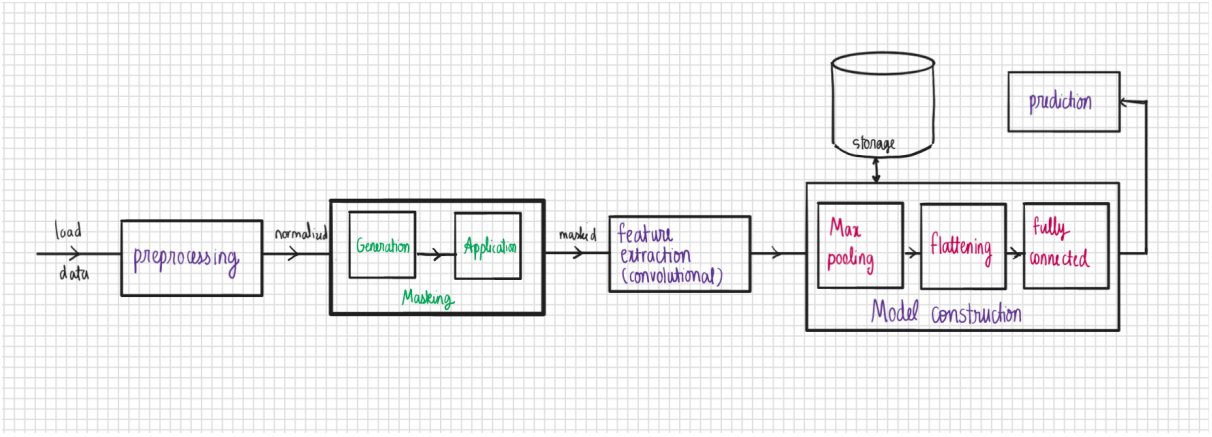
Block Diagrams



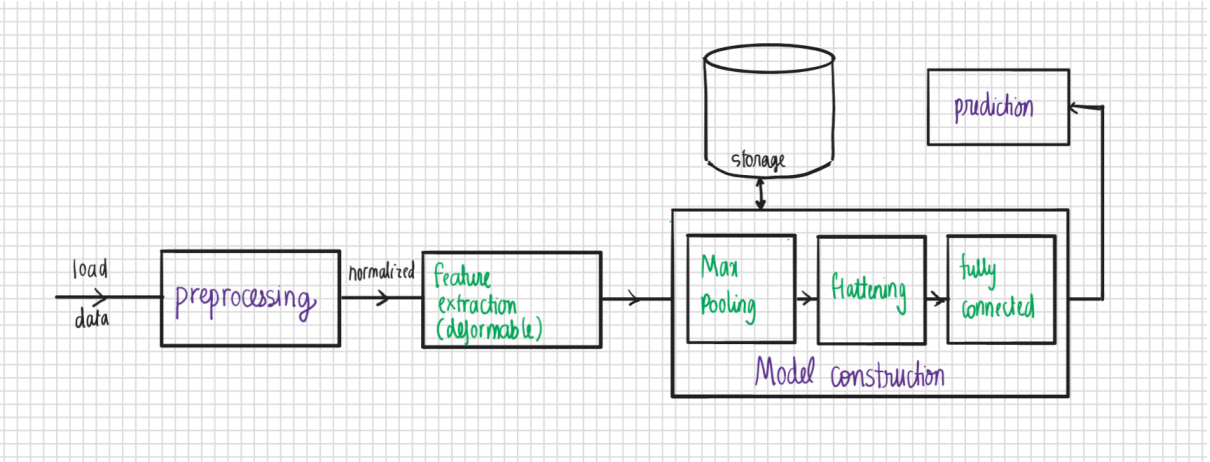
CNN



Masked CNN



Deformable CNN



Algorithms

Libraries used:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers, models
from keras.datasets import fashion_mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import SGD
from keras.utils import to_categorical
import os
```

1. Preprocessing (keras)
    a. Resize all images to a fixed size (128 x 128)
    b. Normalize pixel values between 0 and 1
2. Masking (tf)
    a. Generate a mask to identify regions to focus on
    b. Apply the mask (zero out irrelevant regions and only pass on selected areas)
3. Feature extraction (convolutional)

a. 2 stacked Conv2D layers

b. Each convolutional layer has 32 filters to detect edges, textures, shapes)

c. ReLU activation function is used

```
model.add(layers.Conv2D(32,(3,3),activation="relu", padd
ing='same',input_shape=(128,128,3)))
```

4. Feature extraction (deformable)

   a. 2 stacked DeformableConv2D layers

   b. Each convolutional layer has 32 filters to detect edges, textures, shapes)

   c. ReLU activation function is used

   d. Addition of dilated convolutions to help detect subtle manipulations

```
model.add(layers.DeformableConv2D(32,(3,3),dilation_rate
=2,activation="relu", padding='same',input_shape=(128,12
8,3)))
```

5. Model construction

   a. Max pooling: to reduce spatial dimensions while keeping important features

```
model.add(layers.MaxPooling2D((2,2), strides=(2, 2)))
```

   b. Flattening: to convert 2D feature maps into a 1D vector (to feed into dense layer)

```
model.add(layers.Flatten())
```

   c. Fully connected (dense layers): relu, sigmoid activation to classify the image

```
model.add(layers.Dense(128,activation="relu"))
model.add(layers.Dense(256,activation="relu"))
model.add(layers.Dense(1,activation="sigmoid"))
```

6. Prediction

   a. The final probability (real/fake)

Implementation and Evaluation Metrics

We will be using the OpenForensics Dataset, which in total contains 190k images (both fake and real) split into Train, Test, and Validation sets.

We will be comparing the 3 models on the following metrics: accuracy, precision, recall, F1-score, loss, confusion matrix, detection time, and memory usage.

As per the given timeline, the detailed document on evaluation metrics will be submitted on 01/10/2024, along with 40% of the project implementation.

References

[1] Alben Richards, Kaaviya Varshini, Diviya N et al, “Deep Fake Face Detection using Convolutional Neural Networks”, IEEE, 2023.

[2] Patel, Y., Tanwar, S. et al, “An Improved Dense CNN Architecture for Deepfake Image Detection”, IEEE, 2023.

[3] Shraddha Suratkar, Faruk Kazi et al, “Exposing DeepFakes Using Convolutional Neural Networks and Transfer Learning Approaches”, IEEE, 2020.

[4] Hayat Al-Dmour, Afaf Tareef et al, “Masked Face Detection and Recognition System Based on Deep Learning Algorithms”, Journal of Advances in Information Technology, 2023.

[5] Jifeng Dai, Haozhi Qi et al, “Deformable Convolutional Networks”, Microsoft Research, 2017.

[6] Donnelly, J., Barnett, A.J. et al, “Deformable ProtoPNet: An Interpretable Image Classifier Using Deformable Prototypes”, IEEE.