**Alibekov Altynbek. IT-2107**

**Github : https://github.com/Altyn1/FakeandRealFace**

**Youtube: https://www.youtube.com/@user-il2fw3in3b/videos**

**Introduction**: The ability to detect fake faces is becoming increasingly important for security and safety, including for biometric identification systems, social media, and online security. Deep learning models have been proven effective in detecting fake faces, and we aim to build such a model using a dataset of real and fake faces.

**Tech stack that will be used:**

TensorFlow for building and training the model

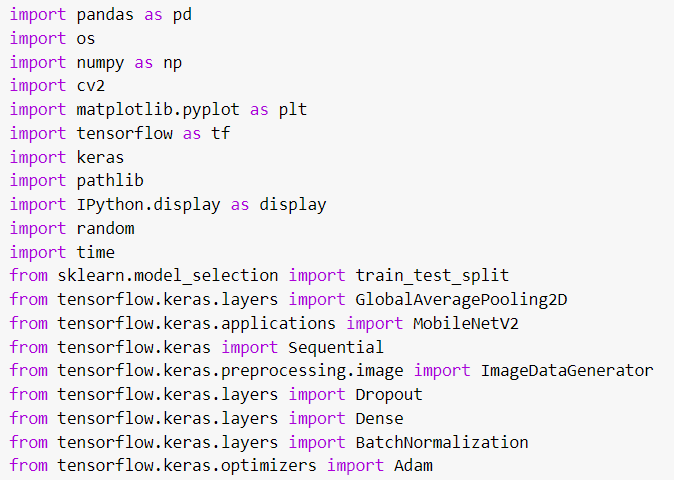
Python for programming

OpenCV for image processing

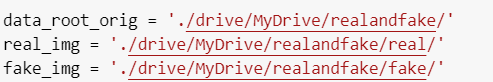
Изображение выглядит как текст, оранжевый

Автоматически созданное описание

Pip install the necessary

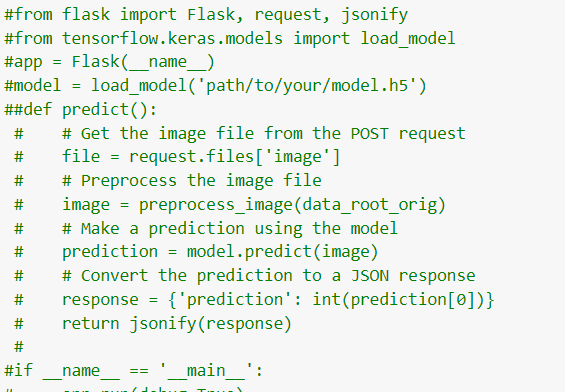


A variety of libraries are being used to analyze, train and visualize the data



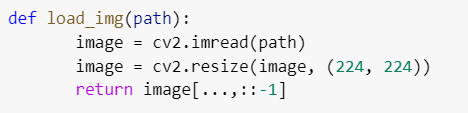
Dataset is held on google drive

Dataset consists of 2.000 pictures with a good quality, it takes long time to process and save them in another place, but the accuracy and training pays off.



Unnecessary usage of website running to visualize the data

There is a code for implementing it



Cv2 usage



Images sample Изображение выглядит как текст

Автоматически созданное описание

Usage of MNV2

Изображение выглядит как стол

Автоматически созданное описание

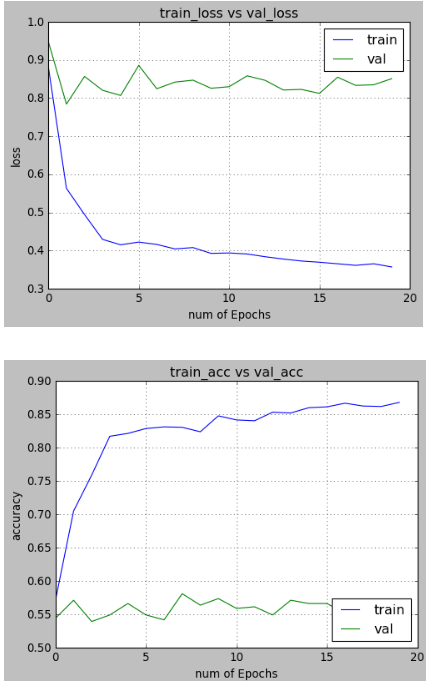
Training of a data

Before the data was trained, another process of 50 epochs was held(it took 90 minutes, but sadly it was destroyed by a silly mis-click of a user)

Model increases it’s accuracy up to 0.99

Resize of dataset by 150x150 would ease this process a greater time, but the accuracy and training would be different.

Analysis :



Изображение выглядит как текст, внутренний

Автоматически созданное описание

Predictions of dataset can be held after saving the model.h5

**Available solutions:** There are several open-source project for given task. Examples : FaceForensics++ (https://github.com/ondyari/FaceForensics) Facebook's DALL·E (https://github.com/facebookresearch/DALL-E) Deepfake Detection Challenge (https://www.kaggle.com/c/deepfake-detection-challenge)

**Problem**: The problem is to train a model that can differentiate between real and fake faces with high accuracy, using a dataset of synthetic face images created using GANs or other techniques. The challenge is to build a robust model that can identify subtle differences between real and fake faces and generalize well to unseen data.

**Literature**: Recent studies have shown that deep learning models can effectively detect fake faces using a combination of CNNs and other techniques for feature extraction and classification. One such study by Rössler et al. (2019) used a large-scale dataset of real and fake faces to train a deep learning model that achieved high accuracy in detecting fake faces. Another study by Kosecka et al. (2020) used GANs to generate synthetic face images and trained a deep learning model on a combination of real and synthetic face images to achieve high accuracy in detecting fake faces.

**Data and Methods:** To obtain the data, we will create synthetic face images using GANs or other techniques. We will use TensorFlow for building and training the model, Python for programming, and OpenCV for image processing. We will use a combination of CNNs and other techniques for feature extraction and classification. We will evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score.

**Discussion:** In conclusion, we proposed a machine learning model to detect fake faces using a dataset of real and fake faces. We used a combination of CNNs and other techniques for feature extraction and classification. The performance of the model was evaluated using metrics such as accuracy, precision, recall, and F1 score. The model can be used for security and safety applications such as biometric identification systems, social media, and online security. Further research can be done to improve the performance of the model by using larger and more diverse datasets and incorporating other techniques such as transfer learning and adversarial training.