Maksims Kazijevs

02/05/2022

COMP6200

**Project 6**

**Problem definition:**

Based on Project 5, I have used the CNN-based model: Xception for multi-label image classification. Originally, author has implemented the model for binary class classification, however I have decided to make it more complex and interesting. Pre-trained model will have additional layers and will be trained with new data. Most parts of the code were changed for my convenience; however, I will still use Xception model that was used by the author in Project 5.

**Design:**

Design will consist of multiple phases: data preparation and processing, model training and testing, and evaluation of the model. I have used tools such as: Python 3.8, Tensorflow, Numpy, Sklearn, Matplotlib and other built-in python packages.

Data preparation and processing:

Collected data will be loaded and corresponding labels will be stored. All images will be decoded, resized to 299x299 resolution, and normalized. Initially, labels will be in String format for a corresponding person, for my problem, I have converted them into one-hot categorical encoding, such that each label is a list, where one row has value of 1 and others are 0 (position of 1 in the list depends on the label of the person). Additionally, I have augmented images for Maksims to balance a data set. Data set was split into training, validation, and testing folds (80/20/20).

Model:

Pre-trained Xception model was loaded, which consists of 132 layers. Since I brought new data, I had to fine-tune the model. For that I had to make last 66 layers learnable (weight and biases will be updated). After Xception model, I will flatten the Xception output matrix and add two fully connected layers. One fully connected layer will have 128 neurons with ReLU activation function and last fully connected layer will have 5 neurons (5 prediction labels) with softmax activation function to calculate probability of a label. I have changed the loss to categorical cross-entropy for multi-label prediction. Callbacks such as ReduceLROnPlateau and ModelCheckpoints are used. ReduceLROnPlateau will decrease the learning rate if validation loss won’t be improving, while ModelCheckpoints will save the model with the best validation loss. Model is being trained for 20 epochs.

Evaluation:

Classification report, accuracy score and confusion matrix are shown along with images with predicted label.

**New data:**

New data was collected from Kaggle, it will feature images for 4 celebrities: Kobe Bryant, Kane Williamson, Maria Sharapova and Ronaldo. Additionally, I have used images from myself to predict 5 different people. Total number of collected data is 477 samples and after augmentation it is 549 samples.

Data set source: <https://www.kaggle.com/datasets/yaswanthgali/sport-celebrity-image-classification?resource=download>

**Execution results, test, and evaluation:**

*Data loading:*

Text

Description automatically generated



*Image augmentation:*

Graphical user interface

Description automatically generated with medium confidence 

*Training data:*

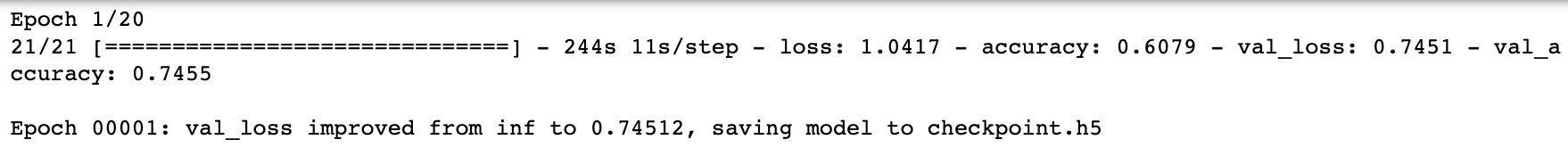
A collage of photos

Description automatically generated with low confidence

*Model loading, parameters, and training:*

 Table

Description automatically generated

 Graphical user interface, text, application

Description automatically generated

*Forward to last epochs…*

*Graphical user interface, text

Description automatically generated*

*Accuracy and loss plots*

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated A picture containing text, receipt, screenshot

Description automatically generated A picture containing chart

Description automatically generated

*Predicted labels and corresponding images*

*Graphical user interface, website

Description automatically generated*

Model has achieved 86.36% testing accuracy.

**Work citation:**

Codingbunnie, CattoxDoggo-Classifier-XCeption, (2020), GitHub repository, <https://github.com/codingbunnie/CattoxDoggo-Classifier-XCeption>

Text

Description automatically generated**Source code:**

Graphical user interface, text, application, email

Description automatically generated Text

Description automatically generated Text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Text

Description automatically generated