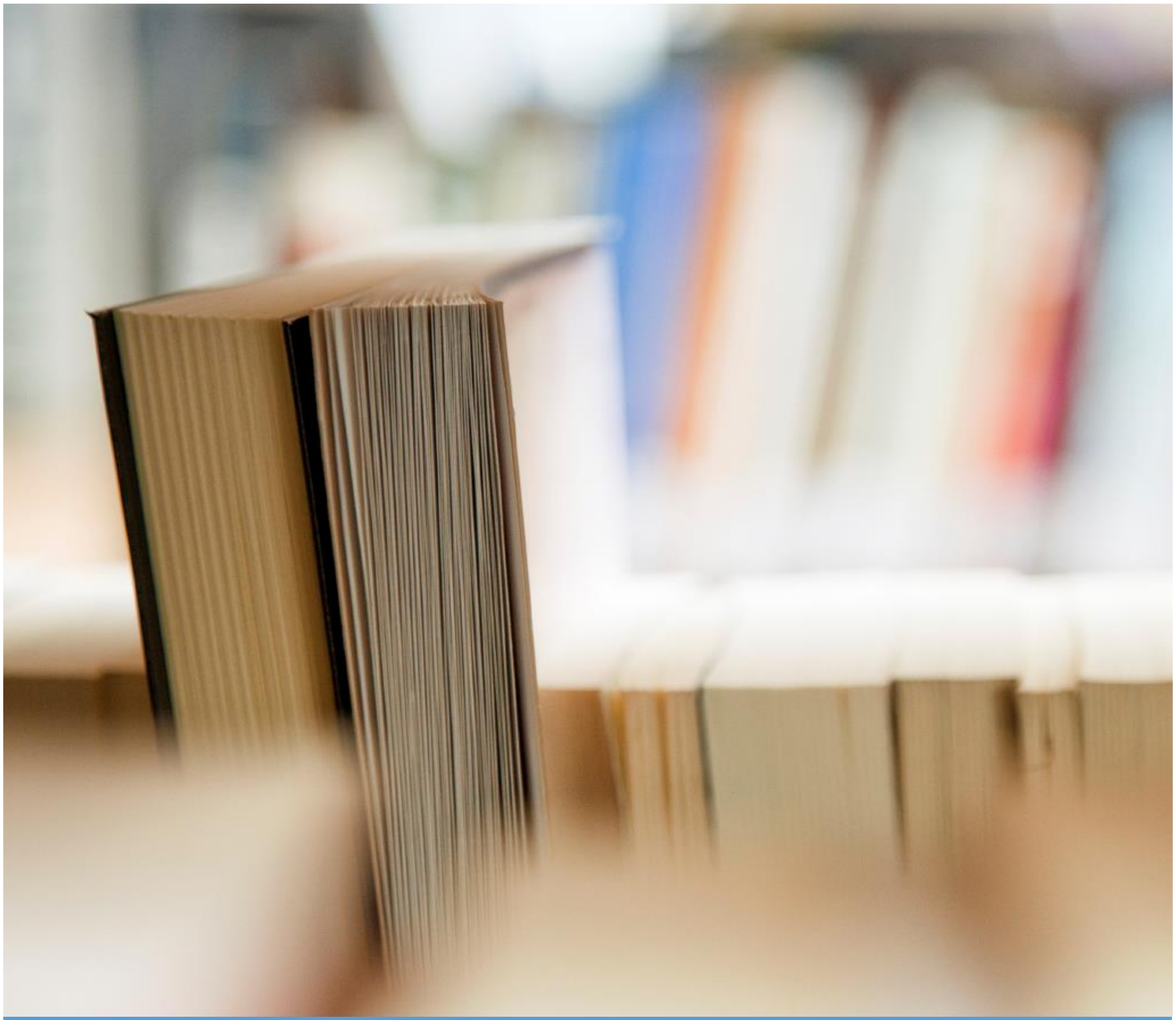


Human Pose Detection Project

1/23/20 Selected Topics in Deep Learning

Introduction

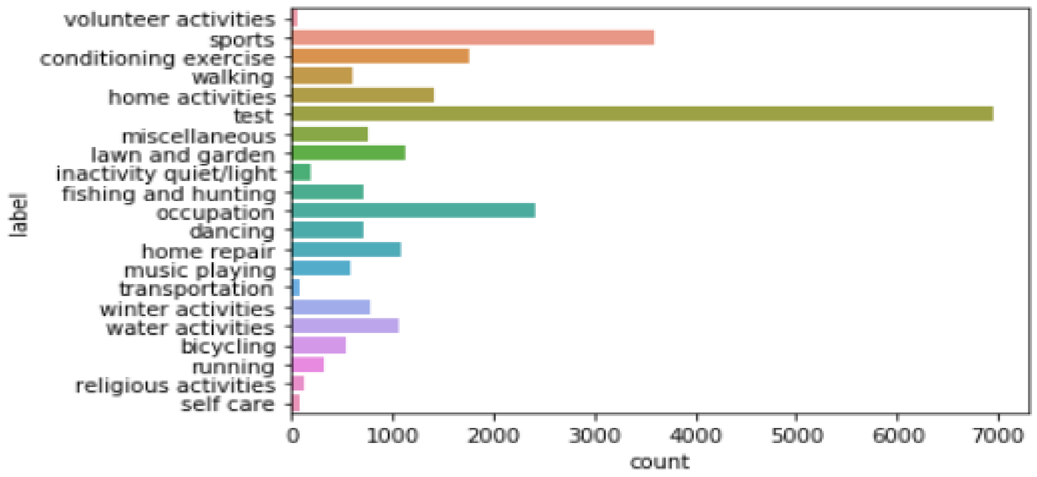
In this project we will develop a deep learning model that estimate the human activity category assigned to each image from MPII Human Pose dataset.

MPII Human Pose dataset is a state of the art benchmark for evaluation of articulated human pose estimation. The dataset includes around **25K images** containing over **40K people** with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities.

Model Summary

We can divide the project into three parts annotation processing, loading images to Google Colab and Model selection so we delivered each step in a separate notebook. We will discuss each part of the project briefly.

- For the **annotation processing** part our target is to extract category name and image name and for the test images it has no category name so we added another category named test for these images(6954). Also it was noted that the classes are not equally distributed in the dataset so we used Stratified sampling for dividing the data into training/Testing/Validation.



- For **Loading Images** part we tried to to read all images at the same time but we faced a computation issue as we were blocked by the RAM size provided by Google Colab so we decided to read then in chunks but RAM also crashed so we finally decided to read chunk of images, resize them to 224*224 and store the resized images in numpy array and delete the unresized images.

- Finally for **Model Selection** part we used VGG16 pretrained model in addition to some built from scratch CNNs and Xception pretrained model and we got almost same results from VGG16 and built from scratch CNNs.

Evaluation Metrics

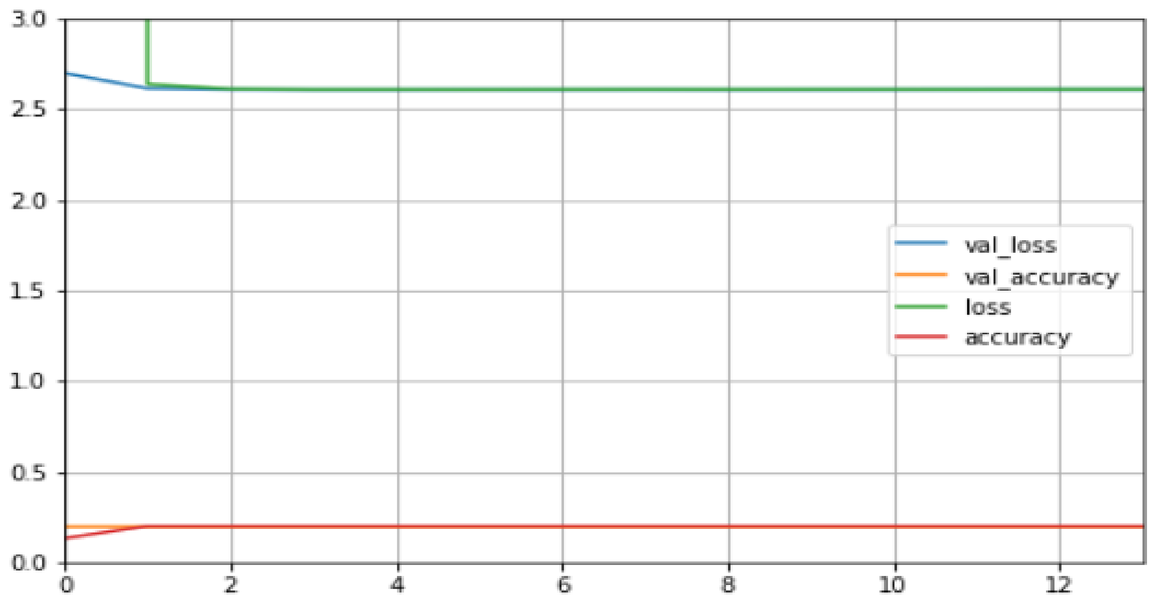
As the dataset is severely imbalanced it isn't useful to use accuracy for performance measurement. F1 Score was the main metric we checked during the different experiments. And it was nearly the same when we used both VGG16 based model and built from scratch CNN.

We measured F1 score using the usual sklearn evaluation metrics after finishing the training part instead of adding this metric during training because f1 score is global metrics while Keras works in batches. As a result, it might be more misleading than helpful.

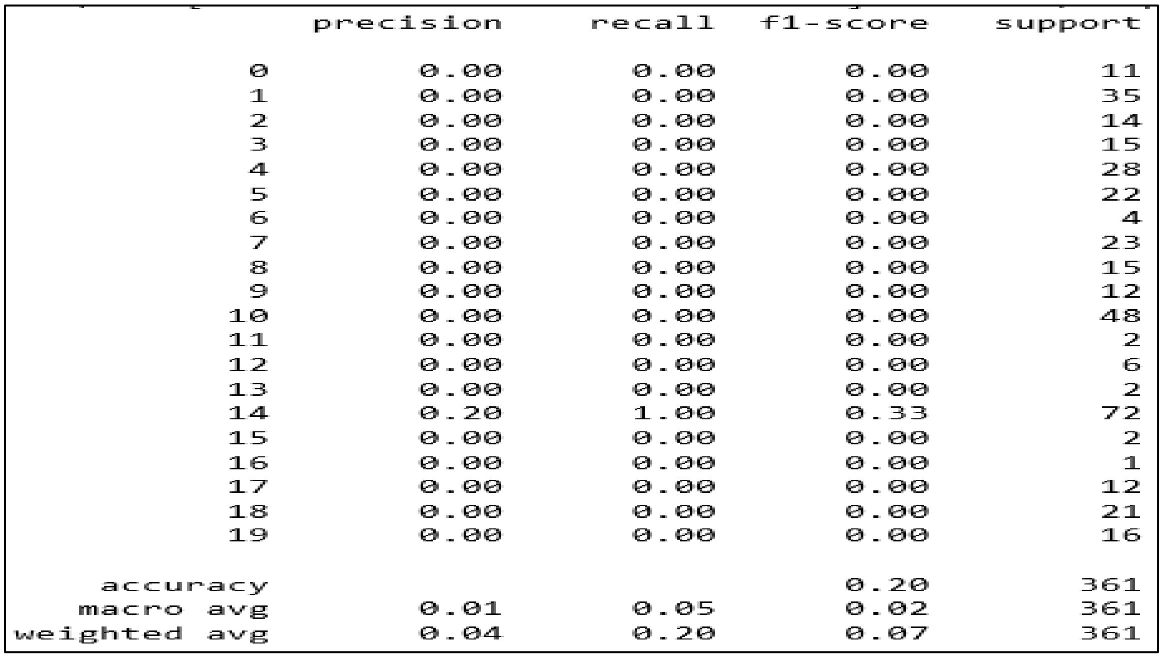
We will present here the results for VGG16 Model and the results for other models are presented in the notebook.

The model that we just removed the top layers and replaced by softmax layer worked slightly better than the model in which we added small convolutional network before softmax layer.

Loss was relatively high as seen below but this is the best we can get to avoid overfitting noting that we tried to increase learning rate as much as we can but learning was very slow in all our experiments.



Also F1 score was only 7% which is very low but compared to the very low number of images(18000 compared to 14M for Imagenet dataset) we can accept that.



```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        11
           1       0.00      0.00      0.00        35
           2       0.00      0.00      0.00        14
           3       0.00      0.00      0.00        15
           4       0.00      0.00      0.00        28
           5       0.00      0.00      0.00        22
           6       0.00      0.00      0.00         4
           7       0.00      0.00      0.00        23
           8       0.00      0.00      0.00        15
           9       0.00      0.00      0.00        12
          10       0.00      0.00      0.00        48
          11       0.00      0.00      0.00         2
          12       0.00      0.00      0.00         6
          13       0.00      0.00      0.00         2
          14       0.20      1.00      0.33        72
          15       0.00      0.00      0.00         2
          16       0.00      0.00      0.00         1
          17       0.00      0.00      0.00        12
          18       0.00      0.00      0.00        21
          19       0.00      0.00      0.00        16

    accuracy                           0.20       361
   macro avg       0.01      0.05      0.02       361
weighted avg       0.04      0.20      0.07       361
```

Limitations and how to scale the model into full dataset.

- First of all the data is too small to output good results specially that number of categories is too much so much more data needed to achieve better results.
- The best model based on the available data may not be the best in case of larger data as we may need to use deeper model like Resnet or Xception for better performance.
- For such relatively small data we faced problems in reading the images because of RAM limitation so we had to resize the images in order to be able to store all of them. Some of the dimensions were reduced to 20% of their original size and as a result we had lost some features that can make the model performance much better. So as a conclusion we need much more powerful machines to process the full dataset which will make us able to use deeper models and a higher training time is needed.
- We thought to use the test images in order to make unsupervised pretraining before working with labelled data but the number of test images is also not that much(less than 7000 images) so we thought that this will consume time without much effect. But generally speaking if we can collect large number of unlabeled images it can help us to achieve better performance especially for the complete dataset.

Computational Effort

VGG best model finished the training phase in 1744.8910219669342 seconds while Xception model training phase took around 1900 seconds with lower performance.

References

https://riptutorial.com/keras/example/32608/transfer-learning-using-keras-and-vgg

https://github.com/selvam85/Cat-Dog-Classifier/blob/master/DNN_using_plain_TF_Cat_vs_Dog_classifier_Kaggle_dataset/Convert%20Images%20to%20Numpy%20array%20and%20save%20in%20h5%20fomat%20v2.1.ipynb

https://datascience.stackexchange.com/questions/45165/how-to-get-accuracy-f1-precision-and-recall-for-a-keras-model