Original Image from https://www.weareworldquant.com/en/thought-leadership/understanding-images-computer-vision-in-flux/

Image Classification using Machine Learning and Deep Learning

Islam Hasabo Follow
May 1, 2020 · 11 min read ★

1 Introduction

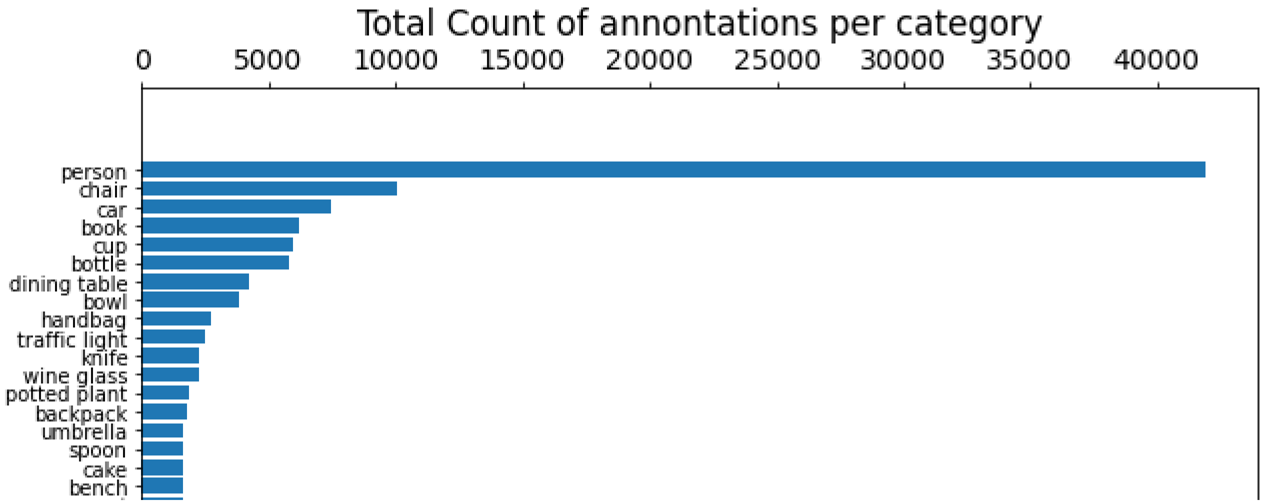
In this project, we will *introduce* one of the core problems in computer vision, which is image classification. It is defined as the task of classifying an image from a fixed set of categories. Many other computer vision challenges such as object detection and segmentation can be reduced to image classification. Throughout this project, we will start by exploring our dataset, then show how to preprocess and prepare the images to be a valid input for our learning algorithms. Finally we will explain relevant and the implemented machine learning techniques for image classification *such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN).*

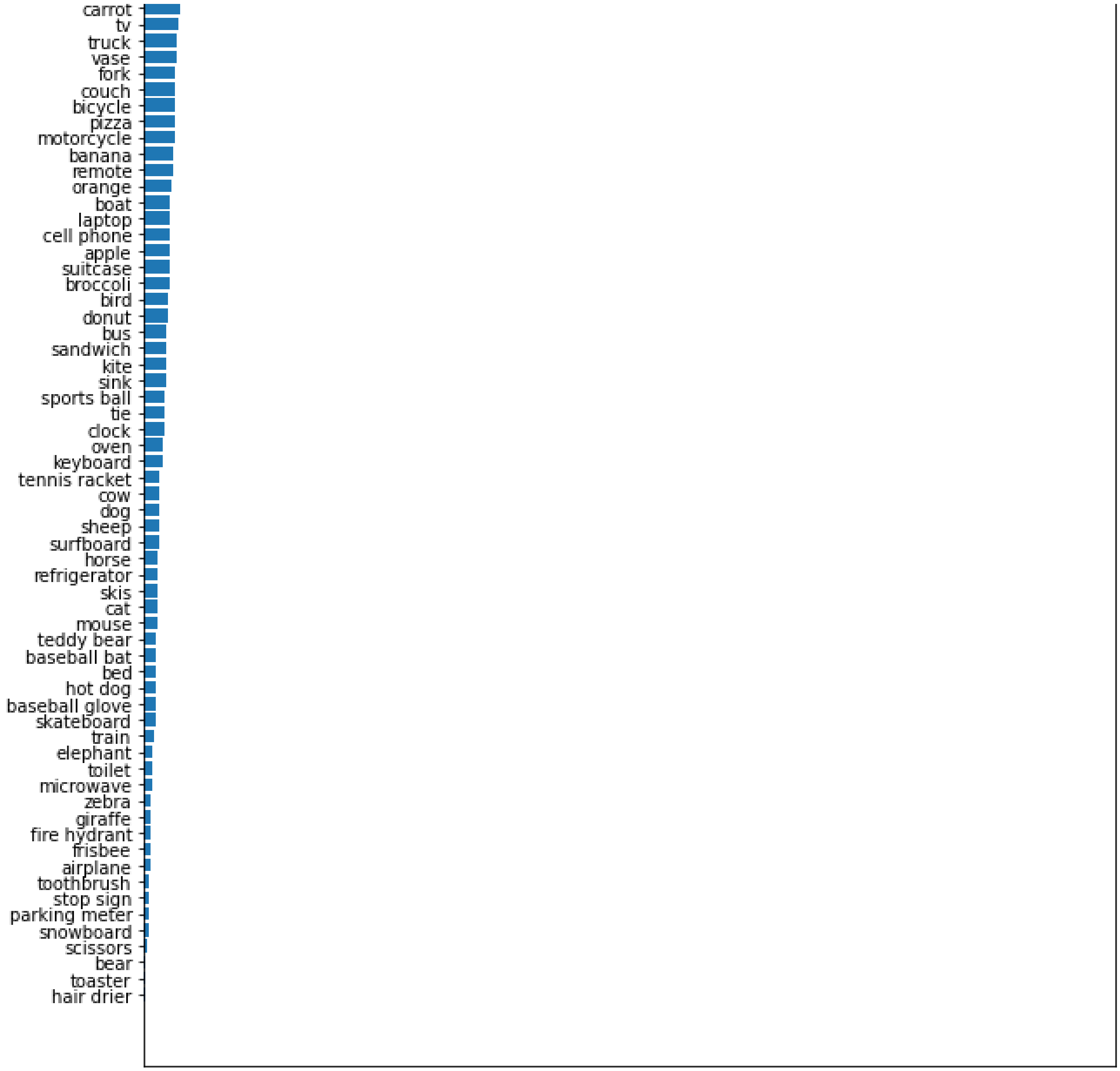
The complete code for this project is available as a Jupyter Notebook on GitHub. This project was a coursework in my masters.

2 Introduction to the dataset

COCO Sponsored by Microsoft is a large-scale object detection, segmentation, and captioning dataset. It contains images, bounding boxes and labels. There are two versions 2014 and 2017 that use the same images but different train, validation and test splits. COCO defines 91 classes but the data only uses 80 classes and some images don’t have annotations. For the purpose of this project, we picked the 2017 validation dataset which is about 5000 images.

Through COCO API, we found out that most images have more than one object with annotations for the same or different category. That led us to make use of this property by cropping the objects using the bounding boxes annotation and considering them as independent images in order to overcome the multi classes in the same image problem so we can feed the models with a single class image which makes it easier to classify. As a result, the number of images multiplied more than 20 times as shown in the chart below.





Due to the computation limitation and even though we have selected the smallest set from COCO, loading all these images wasn't possible. Thus, we had to take half of them prioritizing the ones with low count per class by using the API to get and load the image IDs associated with each category. After that selecting the categories with a number of instances from 150 to 450 which results in 30 classes from 80 that in turn will be used to train , validate and test our models.

3 Literature review of relevant machine learning techniques

Before going through different techniques that can be used for image classification. Let's have an idea about some of the challenges from computer vision perspective which are trivial for a human to perform:

1- Viewpoint variation: Object orientation with respect to a camera.

2- Scale variation: The Same class can exhibit variation in size.

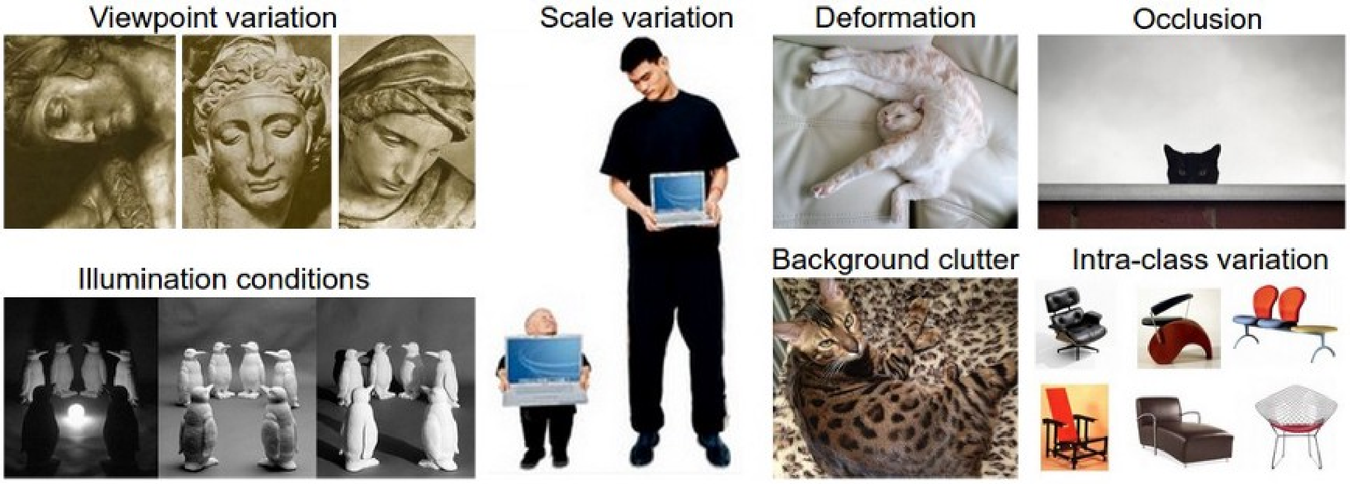
3- Deformation: The nature of the object that it can't be rigid all the time.

4- Occlusion: Sometimes only a small portion of an object could be visible.

5- Illumination conditions: Drastic effect of the illumination on the pixel level.

6- Background clutter: The object of interest to be mixed with its environment

7- Intra-class variation: Broad types of the same class.



original image from https://cs231n.github.io/classification/

In addition, there are about 10,000 to 30,000 different object categories that can be detected. And hence a continuous work and research in this field are trying to find a good image classifier that is invariant to all these variations. By applying and developing machine learning techniques and architectures.

3.1 K-Nearest Neighbor algorithm (KNN)

KNN is a method for classifying objects based on closest training examples in the feature space. The training process for this algorithm only consists of storing feature vectors and labels of the training images. The classification task itself occurs by assigning labels to a testing example by the majority labels of its k nearest neighbors.

The advantages of the KNN algorithm are its simplicity and the ability to deal with multi classes. Nevertheless, a major disadvantage of the KNN algorithm is using all the features equally for similarity computing. This can lead to classification errors, especially when there is only a small subset of features that are useful for classification (JINHO KIM, 2012).

3.2 Support Vector Machine (SVM) algorithm

SVM is a representation of the examples as points in space, mapped so that the instances of the different classes are separated by a dividing plane that maximizes the margin between them. A main advantage of SVM is that it can perform a non-linear classification using the kernel trick. However, a major disadvantage of SVM classification is the limitations in speed and size during both training and testing phases of the algorithm (JINHO KIM, 2012).

3.3 Ensemble Learning algorithm

Ensemble Learning is a technique for aggregating a group of predictors and then predicts the class that gets the most votes. Most popular are bagging and boosting. This way of combining different learning techniques is expected to get better results.

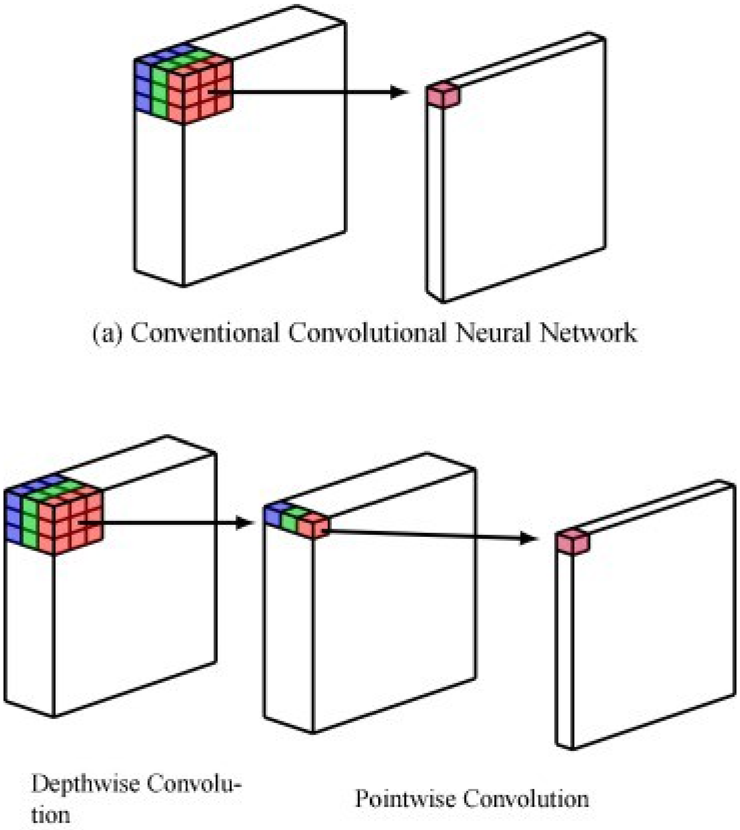
3.4 Multi-layer Perceptron (MLP)

MLP is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. A main advantage of MLP is its capability to learn non-linear models. However, there are major disadvantages such as having a non-convex loss function and tuning for the number of hidden layers and neurons (N. Coskun and T. Yildirim, 2003).

3.5 Convolutional neural networks (CNNs)

CNNs have emerged as the master algorithm in computer vision in recent years, they have managed to achieve superhuman performance on some complex visual tasks with the increase in computational power and one of the outperformer architecture that has been implemented in this project through transfer learning is Xception (Chollet, 2017).

It merges the ideas of other well-known CNN architectures like Inception-v4, GoogLeNet and ResNet, but it replaces the inception modules with a special layer called a depthwise separable convolution (K. Liu, H. Liu, P. K. Chan, T. Liu and S. Pei, 2018). While the standard convolutions simultaneously learn the spatial and cross channel patterns, the separable convolution layer split feature learning over two simpler steps as shown in the figure below. The first step applies a single spatial filter for each input feature map, then the second step looks for cross-channel patterns.


(a) Conventional Convolutional Neural Network

(b) Depthwise Separable Convolutional Neural Network

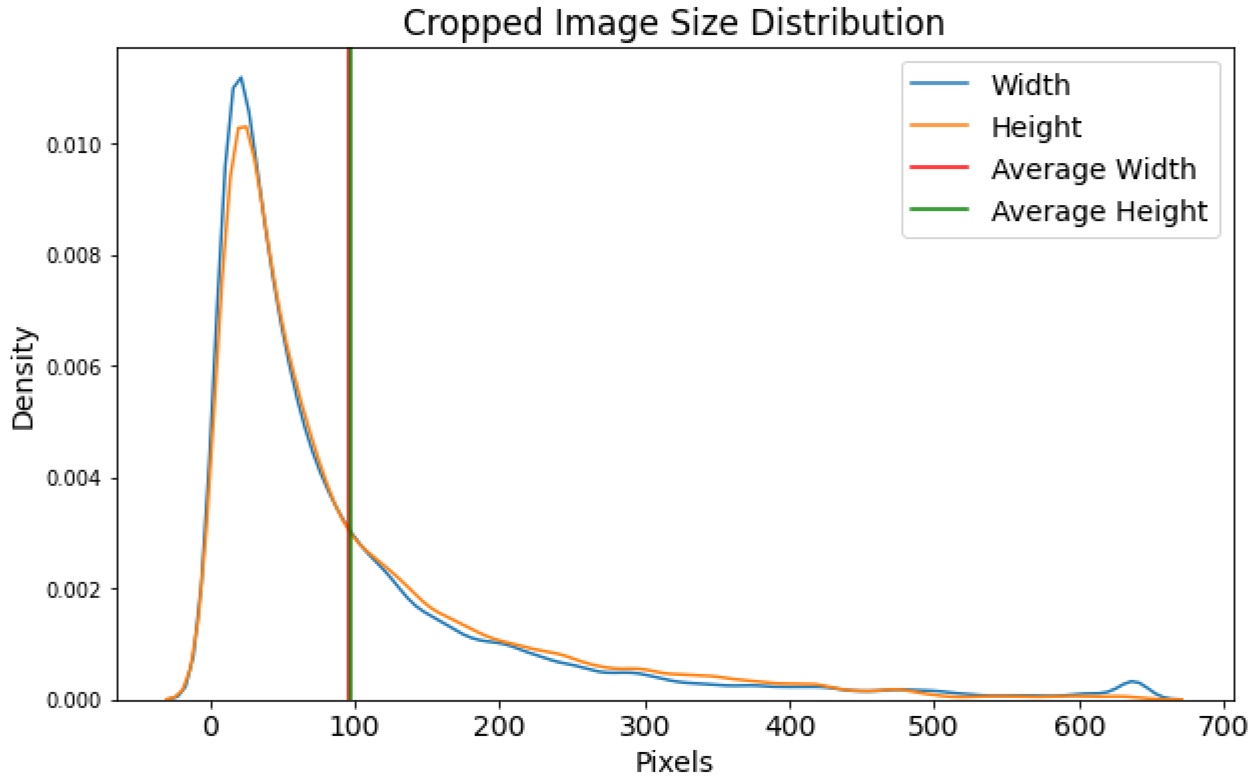
Aside from the better performance, it uses fewer parameters, memory and computations than normal convolutional layers.

4 Image preprocessing and preparation

4.1 Image resizing

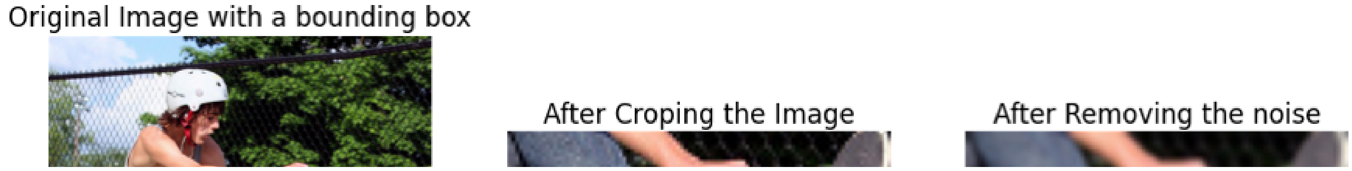
A common problem is that images vary in sizes even without cropping. Images with different height and width are not a valid element to be stacked in an array or input to a machine learning algorithm. Through interpolating pixel color values, the output resized image can introduce a continuous transition. Bicubic interpolation used for this purpose. Although it quite computationally expensive compared to the other interpolation methods, it yields substantially better results. Now, coming to the step of deciding what height and width we can use.

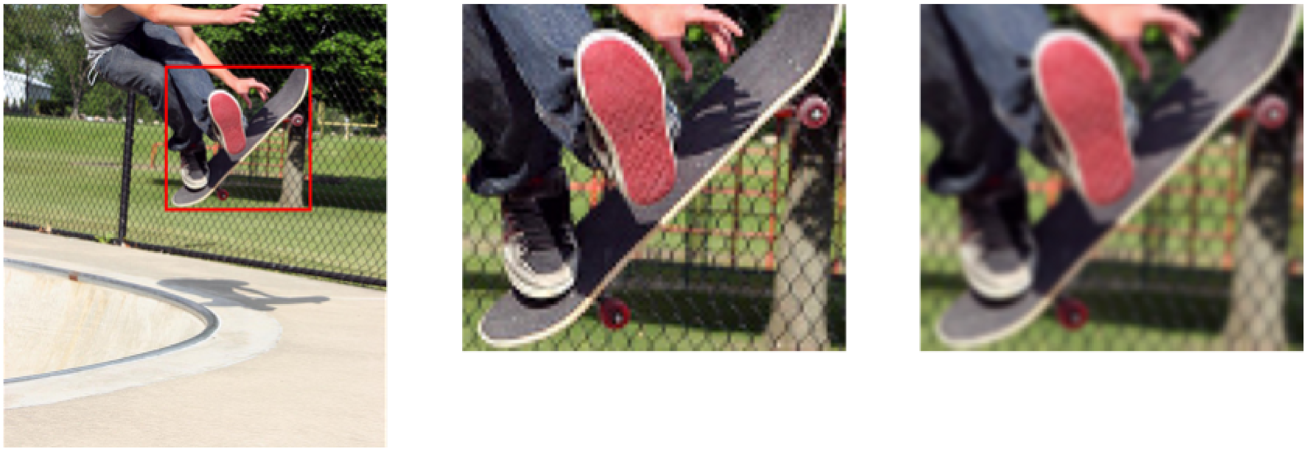
The graph below shows the size distribution of our selected cropped images. Images with width, height or both around 50 pixels are more frequent. However, some images have width or height more than 600 pixels. So, as a compromise we selected the average width and height which is around 100 pixels to be the target size of our resized images.



4.2 Gaussian blur

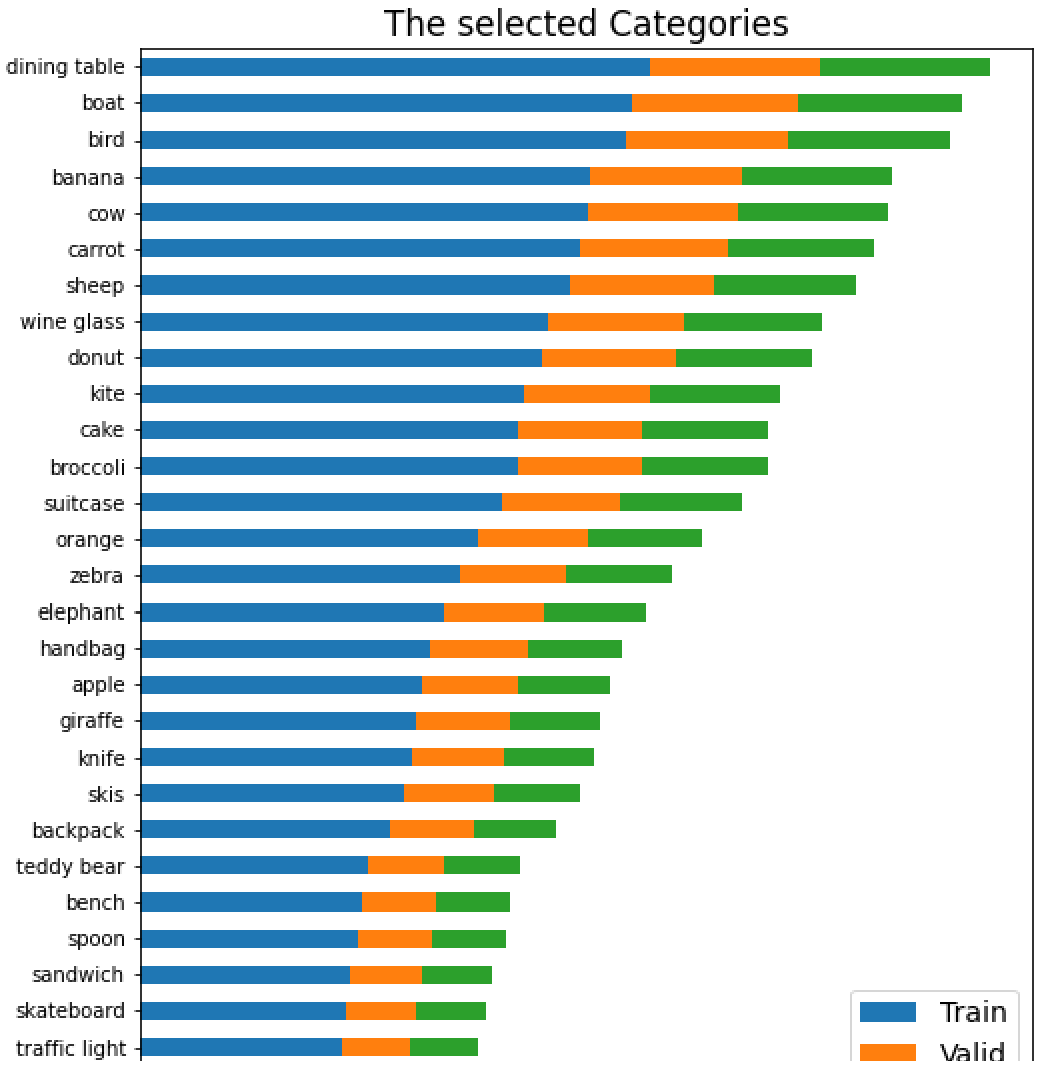
Gaussian blur is one of the widely used process to reduce the noise and enhance image structures at different scales. It is implemented by convolving the image by a gaussian kernel. By trying different kernels, a kernel of size 3 was found suitable for our images. The image below for a skateboard indicates the steps from the original image to resizing then reducing the noise by gaussian blur.

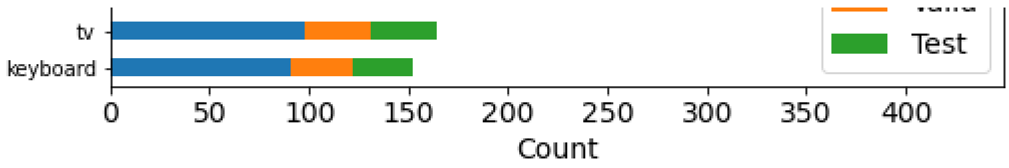




4.3 Data splitting :

Splitting the data was done in two steps. First, stratified splitting with shuffling for all features and labels with percentages 60 for training and 40 for validating and testing. Second, Splitting the 40% using the same way into 20% for validation and 20% for testing as shown on the chart below; the data split distribution for the chosen 30 classes.



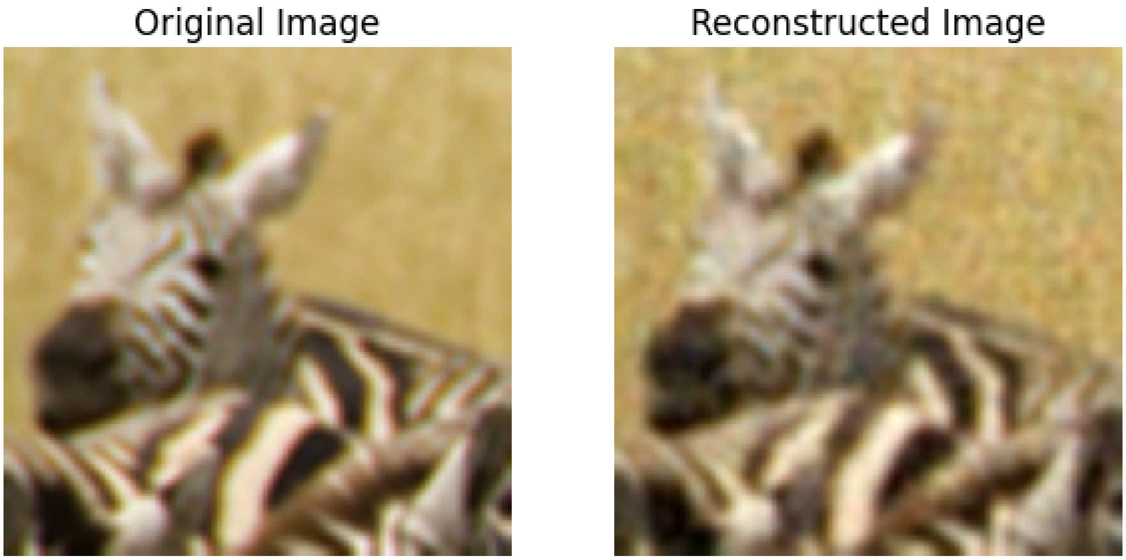


4.4 Data normalization

Simply scaled the features by dividing by 255 to get values ranging from 0 to 1.

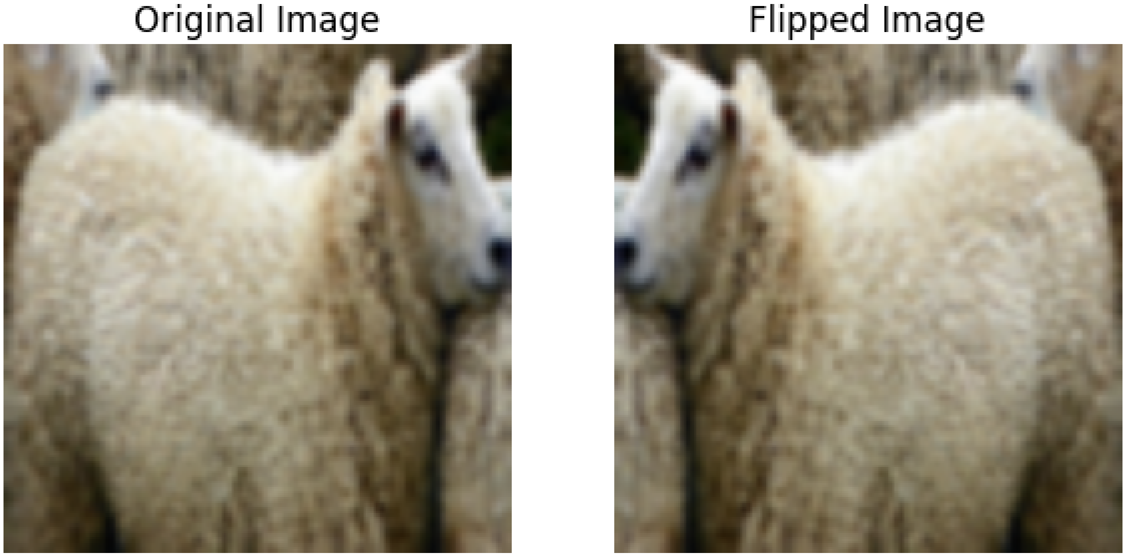
4.5 Principal Component Analysis (PCA)

PCA transformation used for the purpose of dimensionality reduction in order to speed up the training without losing unimportant information by considering keeping 99% of the variances. We even noticed an improved performance from using the non-transformed data. The figure below for a zebra shows that the reconstructed image after inverse PCA didn't change that much from the corresponding original one, while the number of features decreased from 30000 to only 1048.



4.6 Data augmentation

Data augmentation was used only while training our best model taking the advantage of the created TensorFlow datasets that inject the data to the learning algorithm in batches. We selected to randomly do a horizontal flip for our cropped images since it makes more sense than the other data augmentation techniques which can be tried as well in later projects. The figure below for a sheep indicates the difference between the original cropped blurred image and the flipped version of it.



5 Modeling

Six learning algorithms have been implemented with the following notes:

· Google colab with GPU runtime used .

· Validation and testing on the valid set and test set, respectively.

· Since the data is not very skewed, the class weight balancing will not help.

5.1 Stochastic Gradient Descent (SGD) Classifier

SGD classifier used with default hyperparameter hinge loss accounted for linear SVM. It's is a good start because of the advantage of training one instance at a time. It deals with large dataset efficiently and to check the ability to classify the categories linearly.

5.2 Support Vector Machine (SVM) Classifier

SVM classifier used with gaussian kernel and gamma set to auto for the overfitting. Although it takes time for training, this kernel trick depicts the non-linearity.

5.3 K Nearest Neighbors (KNN) Classifier

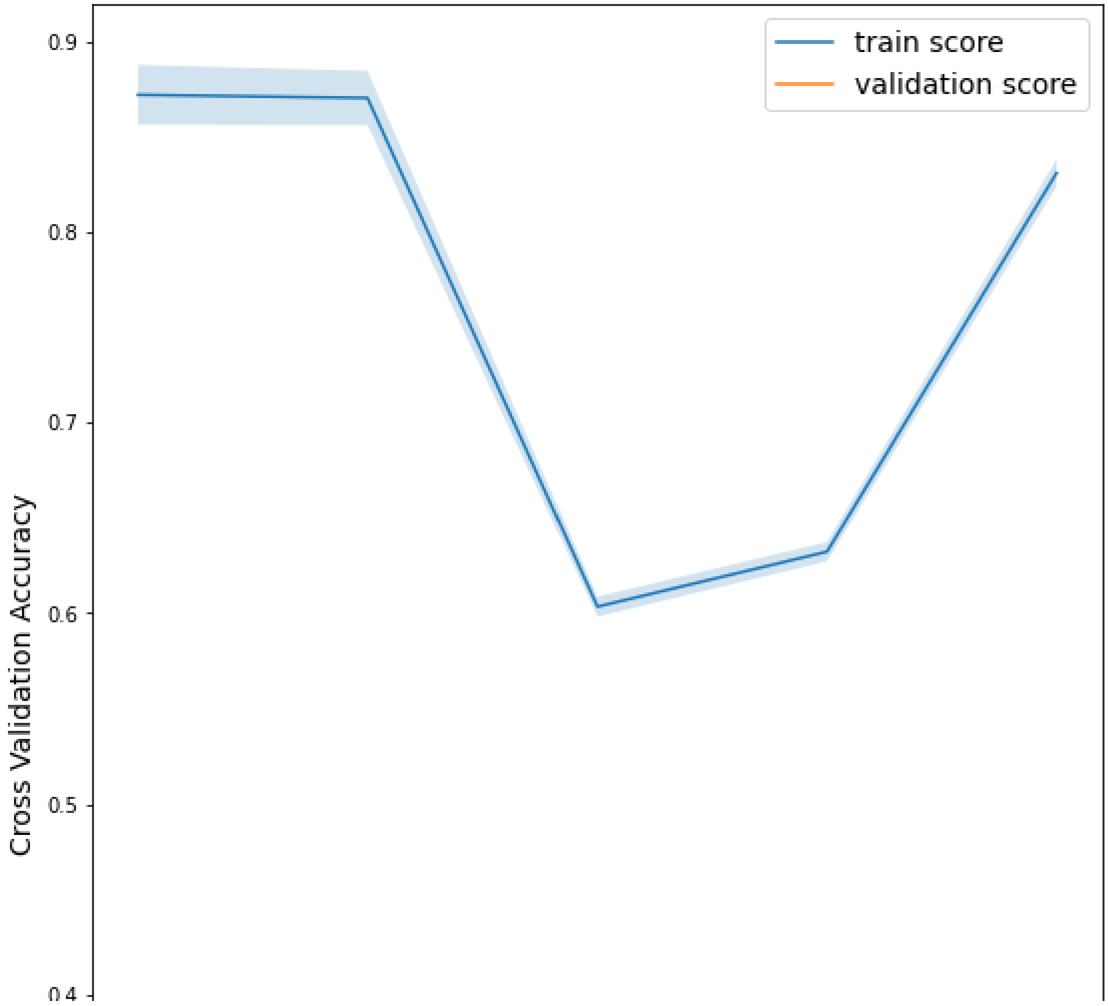
KNN classifier used with manually tuned k to be 11 and distance weights is another simple approach far from the linearity and non- linearity ways.

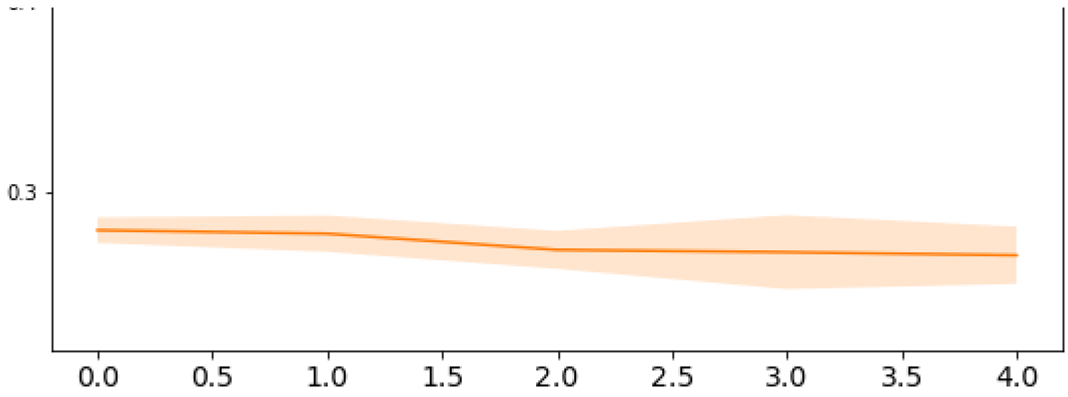
5.4 Voting Classifier

Aggregating the above classifiers with hard voting seeking to see if different learners could be better if they perform together .

5.5 Multi-Layer Perceptrons (MLP) Classifier

Randomly search for the number of hidden layers and neurons with 5-fold cross-validation. The architecture with the best accuracy is input layer with number of features 1048, which is the PCA output + 3 hidden layers each 3000 neurons with relu activation function + 30 units in the output with softmax activation. Considering using the validation set for early stopping during the training which is a way to prevent the overfitting. Adding more data and tuning might improve the performance but not that much.



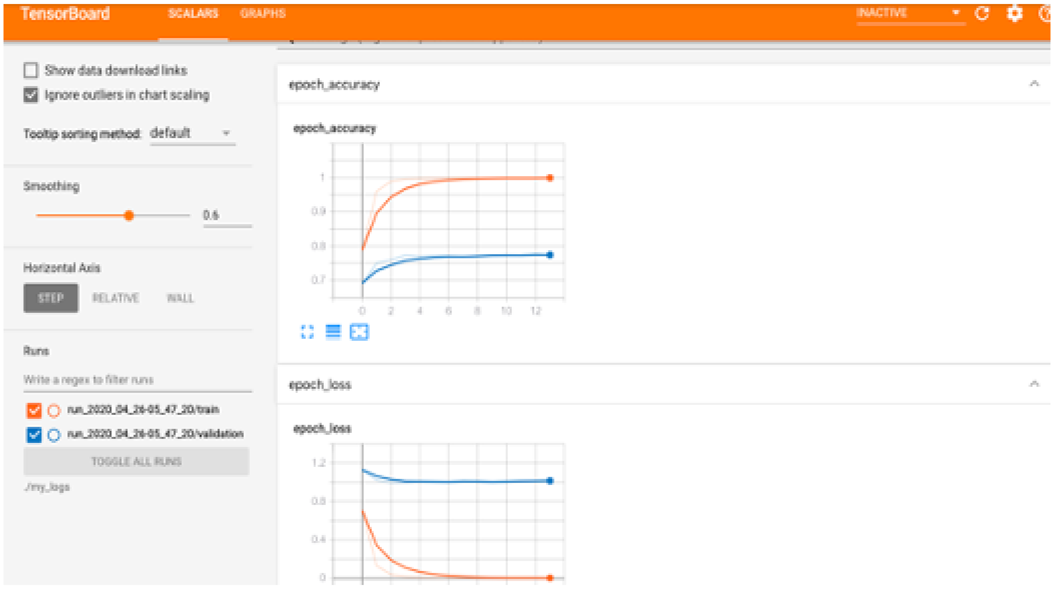


5.6 Transfer Learning using Xception Classifier

The model composed of reused layers with their tuned weights which are the first layers and added layers , average pooling for dimensionality reduction and output layer with 30 units , the number of our classes.

After preprocessing the input by shuffling, augmentation and resizing to match the imagenet weights standard and unnormalize the data to let the preprocess layer handle that , the training done in two steps. First, freeze the reused layers to let the added layer adjust their weights from the initial state. Second, unfreeze the reused ones for fine-tuning all the layers. This model performed the best with testing accuracy 77% which is significantly better than the other learners.

In Terms of bias Variance, as we see from the graph below, this model is overfitting, and hence adding more dropout regularization layers could help. Also, more data required to improve testing accuracy.



□ Show data download links
☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ run_2020_04_26-05_47_20/train
☑ ○ run_2020_04_26-05_47_20/validation

./my_logs

6 Performance

6.1 Accuracy Evaluation comparison

The Chart below compares the performance of our best model against the other baseline techniques on the validation and test sets. Xception outperforms with a margin the other classifiers. The non-linear classifiers such as SVM with Gaussian kernel, Voting and MLP reveal a better performance than the linear ones and KNN. The accuracy on the test set slightly better than on validation set for SVM, Voting and MLP, while the accuracy on validation set is also a little better for the remaining classifiers.

6.2 Time complexity comparison

The Colab GPU was used only MLP and Xception through TensorFlow TF. However, Xception exhibited better utilization due to TF dataset prefetching. Not only the other techniques used the CPU, the Scikit Learn SVM doesn't support the use of all processors as well. In fact, the training for SVM classifier with gaussian kernel is slow compared to SGD, KNN. Also, MLP and Xception training without GPU is very slow.

7 Conclusion

We had an idea about COCO dataset and their annotations that not only can be used for image classification but other computer vision applications as well. We showed the challenges that the computer has to deal with while doing a task like image classification and how image preprocessing help to get better images to train. The PCA ability to reduce the dimensions highly assisted in speeding up training. Data augmentation quite helped to substitute the lack of enough images to train. Although machine learning techniques like SVM didn't give us a good performance compared to a deep learning algorithm like Xception, it was a competitor to MLP in such a way that let us consider first the basic machine learning techniques before going to these computationally expensive deep learning architectures.

8 References

[1] http://cocodataset.org/#home

[2] https://www.immersivelimit.com/tutorials/create-coco-annotations-from-

[3] https://cs231n.github.io/classification/

[4] http://www.wseas.us/e-library/conferences/2012/CambridgeUSA/MATHCC/MATHCC-18.pdf, JINHO KIM, 2012.

[5] N. Coskun and T. Yildirim, "The effects of training algorithms in MLP network on image classification," Proceedings of the International Joint Conference on Neural Networks, 2003., Portland, OR, 2003, pp. 1223–1226 vol.2.

[6] K. Liu, H. Liu, P. K. Chan, T. Liu and S. Pei, "Age Estimation via Fusion of Depthwise Separable Convolutional Neural Networks," *2018 IEEE International Workshop on*

Information Forensics and Security (WIFS), Hong Kong, Hong Kong, 2018, pp. 1–8.

[7] https://arxiv.org/pdf/1610.02357.pdf, Chollet, 2017

[8] https://en.wikipedia.org/wiki/Image_scaling/Bilinear_and_bicubic_algorithms

[9] https://en.wikipedia.org/wiki/Gaussian_blur

Image Classification Computer Vision Machine Learning Deep Learning Tensorflow2