InCity worksheet

|  |  |  |
| --- | --- | --- |
| Date | Major task | Description |
| 17/03 | Image extraction | Looked out the internet to extract bulk images of a certain kind by using Google search. Chrome extension for bulk image downloader can be used. Tried removing incorrectly labelled and cartoon images from the dataset. |
| 18/03 | Model search | Searched for various models present in the internet for binary classification. Figuring out how to apply DL to our own dataset instead of using already available dataset on tensorflow etc. |
| 19/03 | Image augmentation | Looked in the internet for image augmentation and how to do it quickly to increase the data size. There are few applications which would do it for you but it is time consuming. |
| 20/03 | Transfer learning model | Finally instead of using augmentation, as the dataset wasn't in 100Ks used transfer learning with trained weights obtained from a mobilenet\_V2 model. |
| 21/03 | Image extraction | Obtaining images of other categories and apply the model |
| 22/03 | Image extraction | Cleaning data like smaller images |
| 23/03 | Image extraction | Completing the dataset of images and filtering out images by manual process of looking and testing |
| 24/03 | Model evaluation | Using a pre-trained image classification model to check it's accuracy on the given dataset. Mobilenet\_V2 is a neural network whose trained weight we are going to use initially.  Link here: [MobileNet\_V2 results](https://drive.google.com/open?id=1du-TCvErsbRxeSRugzsey-15nNS4R6HpqUQKC61a2-E) |
| 25/03 | Test set evaluation | Looking for documents and articles on adding separate test set or individual test image. Major findings are tensorflow tutorial for transfer learning on tensorflow guide. But keras implementation being better than tensorflow, we need to majorly find it. |
| 26/03 | Dataset cleaning | Removing wrongly placed images, small images, cartoon images from the image dataset |
| 27/03 | Dataset cleaning | Looked for common indian middle class cars through the internet. Added indian origin cars to the vehicle dataset and removed images that had no background noise. |
| 28/03 | Trying to implement a test  Set | Along with the train-valid set, I looked for tutorials on the internet to introduce the test set into the classifier. But the implementation is difficult in tensorflow |
| 29/03 | Trying to implement a test  Set | Tried implementation in keras to figure out the test set, but wasn’t able to do it. FINALLY USING PYTORCH WE GET TEST-TRAIN-VAL.(03/04) |
| 30/03 | Types of image  Classification | Looked for various types of image classification, their application and limitations. Also various types of systems providing deep- learning applications. |
| 31/03 | Checking dataset imbalance | Putting forth a function to calculate the number of images in each category, also get a function for augmentation. |
| 1/04 | Realizing the problem with  Binary classification | Binary classification gives depreciative results for a single class say “Cat vs non-cat” type. Hence multiclass classification is a better approach. To solve the additional prob. With many animals in single pic built a multilabel classification. |
| 2/04 | Building Multilabel classifier  using TF 1.0.X | The classifier gave an accuracy of 77% using a normal Con-2d model, the next task is to figure out how to apply transfer learning to it. |
| 3/04 | Using PyTorch | Given the simpler application of PyTorch, formulated a multiclass model for classification based on the following:  <https://medium.com/@polm23/thanks-for-the-article-32d581f31061>  <https://github.com/bentrevett/pytorch-image-classification> |
| 4/04 | Model compilation based  On the structure given in  Classification dictionary | A compilation of all the image dataset and training the models separately for different categories. Then calculated the test scores. Started working on object detection by following a few tutorials on ‘OBJECT DETECTION VIA TF’ present on medium. |
| 5/04 | Object detection | Started with making the dataset for the object detection, followed the steps given in tutorials with smaller train set.  <https://towardsdatascience.com/detailed-tutorial-build-your-custom-real-time-object-detector-5ade1017fd2d>  <https://www.dlology.com/blog/how-to-train-an-object-detection-model-easy-for-free/>  Faced errors belonging to these:   1. .pbtxt file, can’t make changes to it depending on the model. 2. Model needs a very high version of python and a low version of TF, which wasn’t known or given. 3. The dataset structure is different for most of the tutorial. |
| 6/04 | Data Collection | Looked out the internet to extract bulk images of mobile phones and ACs by using Google search. Chrome extension for bulk image downloader can be used. |
| 7/04 | Building model | Built a model around the newer datasets using binary classification and trained in the same way using transfer learning. |
| 14/04 | Built entire dataset | Built dataset for object detection:   1. Crop all the images to a desired square, preferred is 300\*300. 2. Then label the cropped images by giving coordinates of the 4 end points and giving the label name. 3. Save the given xml file containing a label with the same name as image file. |
| 15/04 | Applied keras object detection | Tried applying keras object detection to the dataset which resulted in errors as the implementation isn’t clear. |
| 16/04 | Application on cows. | Used the following tutorial to apply tf object detection api.  <https://www.dlology.com/blog/how-to-train-an-object-detection-model-easy-for-free/>  <https://colab.research.google.com/github/Tony607/object_detection_demo/blob/master/tensorflow_object_detection_training_colab.ipynb>  <https://github.com/Tony607/object_detection_demo>  Created dataset for cows and fork the his repository and made needed changes. |
| 17/04 | Applied model on cows | Applied Object detection on cows using TF OBJ DET api, the loss was around ~1.2. The algorithm detected the presence of animals accurately.  But resulted in more bounding boxes than one for a single animal. According to my understanding that is mostly because of the different poses of an animal. |
| 18/04 | Application for animals  category | Created a mixed dataset of all the types of animals, labelled and cropped the dataset as required (standard is 800\*600). Forked the git repo given above, deleted the already present data and replaced it with our data. Then made necessary changes in the colab notebook. |
| 19/04 | Object Detection | Applied object detection to all animals with all the categories, with a loss of 1.2(NMAE). |
| 20/04 | Changes into the score | Included a dictionary to calculate the probability score of each category. But couldn’t upload it as a CSV with all the details regarding the predicted category. |
| 21/04 | Inference from the  Present algo | The code ran well in the following cases:   1. Identifying an image belonging to any particular categories. 2. Identifying an image belonging to none of the categories. 3. Identifying two or more instances of the same category in an image. |
| 22/04 | Inference from the  Present algo | The improvements needed:   1. Identification of image with animals of two or more categories in a single image. 2. Improvement in the code to convert probability score into dictionary followed by CSV. 3. More information on the bounding box based on the different poses of the same category. |
| 23/04 | Algorithm for problem  Related to classification | Looked into the internet for models that could help in classification of “A VS not A”. The models like tensorflow Obj Det. API is one of them. |
| 24/04 | Compilation of result | Compiled the result for all the categories, not in the category, many belonging to a single category, many belonging to different categories based on the probabilistic score. |
| 25/04 | Model weight and  label map | Downloaded and saved the trained weight and label-map for the Obj Det. model. |
| 26/04 | Working of fine tuning  Object detection algo | Changed the hyperparameters in Object Detection algorithm so that, the MSE error while model validation is reduced and there is a limit on the number of bounding boxes. |
| 27/04 | Made improvements in  Dataset for proper testing | Tested object detection model on images belonging to animal category:Individual, multiple of same animals and mixed category. Also tried using the algorithm on few images belonging to no c |
| 28/04 | Presented the model  Working in situ | Presented the object detection model for animals. |
| 29/04 | Read about model to help  With none\_class | The major challenge was to include a none class in image classification, there isn’t any standard algorithm for classification of categories as dog vs non-dog. Few techniques like bounds on probability score, sigmoid activation and binary classification at each end neuron have been worked upon |
| 30/04 | About TensorFlow api | Tensorflow provides an API for object detection that can be used to make apps for mobile phones. These services and their charges are mentioned in it’s website. |
| 03/05 | Started working on Dripto’s  Idea of using new layer of  classifier | Another interesting approach can be using an extra layer of classifier at the end of image classification. The main objective of this classifier will be to test if the image is in one of the classes(1) or not in any(0). Classifiers like random forest classification, logistic classification, decision trees or Gradient boosting can be used(XG Boost). |
|
|
| 04/05 | Read a few medium articles  About same | Read a few articles and blogs about better algorithm for the same and the idea of implementation. |
| 05/05 | Implementation of app and its  Similarity to the problem  At hand | The backend in which an app works is mostly inspired by the algorithm, given the fact that there are many instances of object detection application in day to day life like OCR detection, face detection, number plate detection.The idea of implementation could be inspired from an app. |
| 06/05 | Read keras documentation  For datagenerator deeply | Another major challenge was to test the image classification model on random images belonging to the user, found the same in Keras documentation for data generator:  <https://keras.io/api/preprocessing/image/> |
| 07/05 | Improvement in Classification  model | Tested the classification model without none for separate images to check the working. |
| 08/05 | Output for TFlite and  TFlite quantized file | A section on these has been included (as a comment) in the image classification algorithm, these files are downloaded after the model is ran. |
| 09/05 | Infp about these types of  file | TFlite and TFlite quantized files are used to make apps. Whenever an app building process takes place, please refer to the section: “**TF Lite files”** and copy paste the content given in the description in the code cells below. |
| 10/05 | Building algo for testing  Single image in image  classification | Added an algorithm for testing based on the reading of keras documentation. |
| 11/05 | Improving the above for  Bunch of images | Taken forward the idea of testing a single image to testing from a directory, a directory on google drive that contains the image you want to test, with an idea to use a none image folder for training the separate classifier |
| 12/05 | Making dataset for  None\_class classifier | Made an image dataset with 40-50 images belonging to none category like humans, empty road, water pipes, shops, posters etc. |
| 13/05 | Building another algorithm  Using RandomForest  classification | 1. Applied the image detection algorithm on the none\_class folder and recorded the score for each image(0).  2. Merged it with scores of images in a category(1) such that all classes are included equally.  3. Trained a random forest classifier on these images to get an f1 score and accuracy |
| 14/05 | Evaluating for house | Tested a random forest classifier on house images to get an f1 score 0f 0.84 and accuracy 77%. |
| 15/05 | Finalising the model with  The necessary input | Defined the model structure and parameters for this mixed image classification and none classifier. |
| 17/05 | Some EDA on obtained  Dataframe with images from  None category | Tried using the model on the test set, the output function is yet to be defined, that results in final output of the model. |
|
| 18/05 | Model ready | The basic structure of the model is ready. |
| 19/05 | Improving none\_class  Classifier algo with XGB | Read more about XG boost for image classification and its impact a faster neural network, and changed the final classifier algorithm to XGB and the corresponding accuracy was 94% and f1 score was 0.92 |
| 20/05 | Model presentation | Presented the separate none model, but the actual results where really bad and the model was biased toward the first category only. |
| 21/05 | Model Debugging & Newer  Ideas though stack  Overflow and research  papers | Read a few more threads on stack exchange about none\_class classification, follow the same here:<https://stackoverflow.com/questions/43578715/how-best-to-deal-with-none-of-the-above-in-image-classification>  <https://arxiv.org/abs/1511.03719> |
| 23/05 | Went with a naive idea of  Including a none\_class  With as many images. | A very naive but unpredictable approach is to add a none\_class in the image dataset, although this approach isn’t the one used mostly because none isn’t a distinct class but it generally solves the problem and is simpler in implementation. |
| 24/05 | Made the model | Made a model with none\_class for all categories of images, and added a testing section for it. |
| 26/05 | Debugging the separate  none\_model | A very key part of testing the model on a random image is resizing it to the size on which it was trained, I had skipped that part in the separate model hence all the results on the test set were garbage and biased, changed the same here. |
| 27/05 | Presented the model | Made the necessary changes and improved the final testing algorithm for both the models. |
| 30/05 | Tuning the model parameters | Made changes in a few probability score for final classification |
| 6/05 | Making dataset for  separate none\_classmodel | Both the none included and separate none require different dataset:  The address for separate none is:<https://drive.google.com/drive/folders/1DvZ9jK48HwuT9J8eKbjQ0YvkgHMsDljG?usp=sharing>  Open the link and add it to your drive. For more details look the the model guide presentation |
| 7/05 | Made ppt fore model guide  book | The powerpoint contains all the necessary information related to the model and steps to run it as well. |
| 8/05 | Final tweaking in the  parameters | Made a few changes to the threshold for probability score for final classification such the model doesn’t under perform because of them. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |