Deep learning | Image Classification

Contents

Main Objectives:

This report focuses on using CNN algorithm to classify different images by using machine equipped by CPU processor.

The aims of this report are:

1-Classifing the images by using different sequential CNN models.

2- Studying the effect of tuning different hyperparameter on models performance in order to determine which set of parameters have noticeable impact on model behavior.

This report can benefit People who are interested in image classification domain as it an attempt to quantify the relationship between different sets of hyperparameters and models performance.

Data:

Source of data set is data.mendeley site which specialized in data science dataset, find link below to download the dataset

https://data.mendeley.com/datasets/4drtyfjtfy/1

About Dataset:

This dataset consists of 1125 outdoor colorful images for different weather conditions (Shine, Cloudy, Rain and sunrise), their size varies from one image to other and they are proper balanced meaning that number of images for different weather classes are somehow similar.

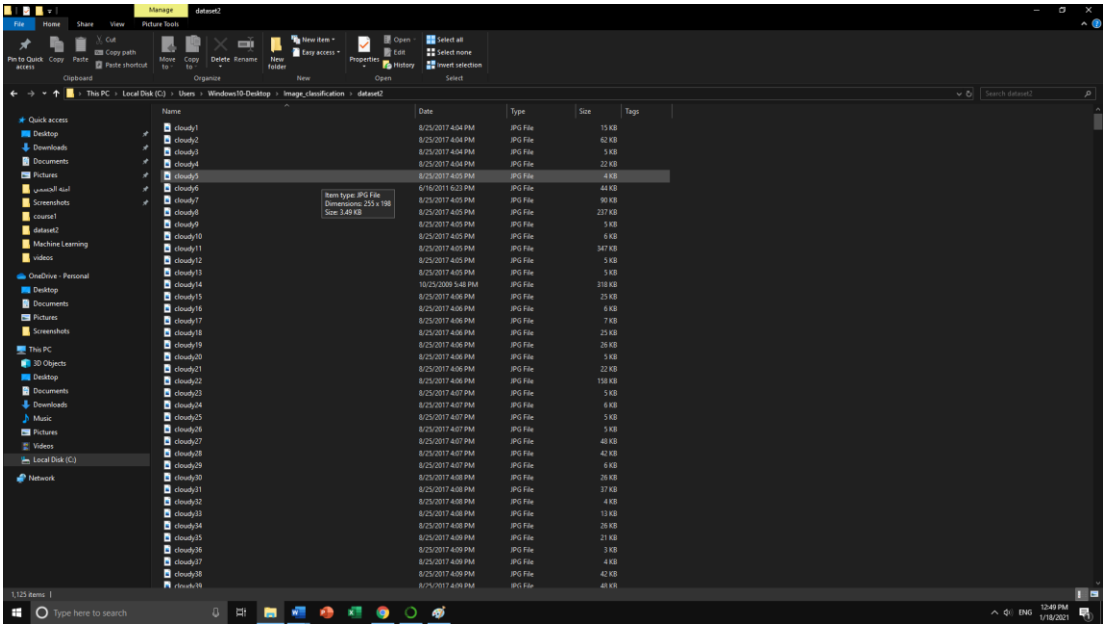
Dataset Attribute:

Image: numpy array (shape (250,250,3)

Class_des: object (values: Shine, Sunrise, Cloudy, Rain)

Class_id: integer (values:0,1,2,3)

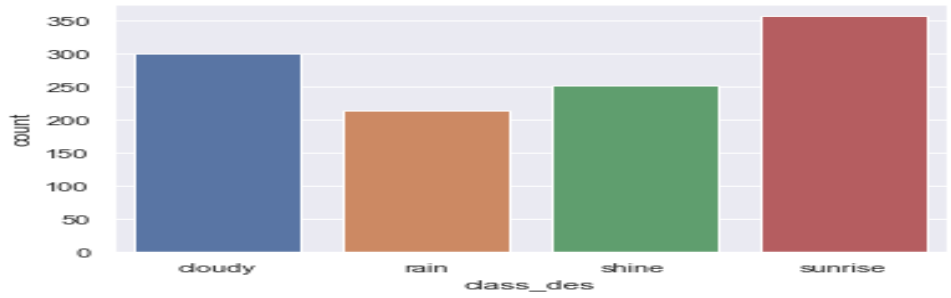
Data comes in form of images file (JPG) and each image name represent the corresponding weather status (screenshot for images files) as part of data features engineering class description and class_id will be extracted.



Sample of images:

Dataset Analysis:

This analysis aims to study dataset distribution and to assure data is balanced, as hown below data is balanced (see the below figure).



Data Exploration:

In this section we will explain in brief the set of actions which been taken to prepare the data for classification model:

1- Read the image file from image directory
2- Convert the image file to array
3- Resize the images to (250,250,3)
4- Extract the class description and class number for each image
5- Rearrange randomly the samples to make sure samples are not ordered by class (When images are loaded from net they are sorted by name so either to resort them in the hosted directly by using other features (For example size), Or to resort them randomly by using np.random.shuffle() method.
6- Distribute the data between training and validation sets (900 samples have been dedicated for training and 225 for validation).
7- Convert classes to categorical by using keras.utils.to_categorical() method.

Model Training:

Different models have been created by tuning different hyperparameters like (number of epochs, number of conv2d layers, type of activation function, type of optimizers, size of kernel..etc)

First Model:

This model served as a bassline for other models, it consists from (conv2d layers , relu activation function, RMSprop optimizer, kernel size of (5,5) , MaxPooling 2D size(2,2).....(See below figure)

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 125, 125, 250)     19000
_________________________________________________________________
activation (Activation)      (None, 125, 125, 250)     0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 61, 61, 250)       1562750
_________________________________________________________________
activation_1 (Activation)    (None, 61, 61, 250)       0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 30, 30, 250)       0
_________________________________________________________________
dropout (Dropout)            (None, 30, 30, 250)       0
_________________________________________________________________
flatten (Flatten)            (None, 225000)            0
_________________________________________________________________
dense (Dense)                (None, 512)               115200512
_________________________________________________________________
activation_2 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 2052
_________________________________________________________________
activation_3 (Activation)    (None, 4)                 0
=================================================================
Total params: 116,784,314
Trainable params: 116,784,314
Non-trainable params: 0
_________________________________________________________________
```

The best accuracy achieved by using validation dataset is 81,78% at epoch 7

Second Model

One more conv2D layer has been added to first model

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_2 (Conv2D)            (None, 125, 125, 250)     19000
_________________________________________________________________
activation_4 (Activation)    (None, 125, 125, 250)     0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 61, 61, 250)       1562750
_________________________________________________________________
activation_5 (Activation)    (None, 61, 61, 250)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 29, 29, 250)       1562750
_________________________________________________________________
activation_6 (Activation)    (None, 29, 29, 250)       0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 250)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 14, 14, 250)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 49000)             0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               25088512
_________________________________________________________________
activation_7 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 2052
_________________________________________________________________
activation_8 (Activation)    (None, 4)                 0
=================================================================
Total params: 28,235,064
Trainable params: 28,235,064
Non-trainable params: 0
```

The best accuracy achieved by using validation dataset is 80.44% at epoch 9

Third Model:

Changing the activation function in model_1 to leaky Relu

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 125, 125, 250)     19000
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 125, 125, 250)     0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 61, 61, 250)       1562750
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 61, 61, 250)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 30, 30, 250)       0
_________________________________________________________________
dropout_6 (Dropout)          (None, 30, 30, 250)       0
_________________________________________________________________
flatten_3 (Flatten)          (None, 225000)            0
_________________________________________________________________
dense_6 (Dense)              (None, 512)               115200512
_________________________________________________________________
leaky_re_lu_5 (LeakyReLU)    (None, 512)               0
_________________________________________________________________
dropout_7 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_7 (Dense)              (None, 4)                 2052
_________________________________________________________________
activation_11 (Activation)   (None, 4)                 0
=================================================================
Total params: 116,784,314
Trainable params: 116,784,314
Non-trainable params: 0
_________________________________________________________________
```

The best accuracy achieved by using validation dataset is 75.11% at epoch 5

Fourth model:

Reducing the kernal size to (3,3)

```
Model: "sequential_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_10 (Conv2D)           (None, 125, 125, 250)     7000
_________________________________________________________________
activation_12 (Activation)   (None, 125, 125, 250)     0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 62, 62, 250)       562750
_________________________________________________________________
activation_13 (Activation)   (None, 62, 62, 250)       0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 31, 31, 250)       0
_________________________________________________________________
dropout_8 (Dropout)          (None, 31, 31, 250)       0
_________________________________________________________________
flatten_4 (Flatten)          (None, 240250)            0
_________________________________________________________________
dense_8 (Dense)              (None, 512)               123008512
_________________________________________________________________
activation_14 (Activation)   (None, 512)               0
_________________________________________________________________
dropout_9 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_9 (Dense)              (None, 4)                 2052
_________________________________________________________________
activation_15 (Activation)   (None, 4)                 0
=================================================================
Total params: 123,580,314
Trainable params: 123,580,314
Non-trainable params: 0
_________________________________________________________________
```

The best accuracy achieved by using validation dataset is 85.78% at epoch 7 (it was run for only 7 epochs)

Fifth Model:

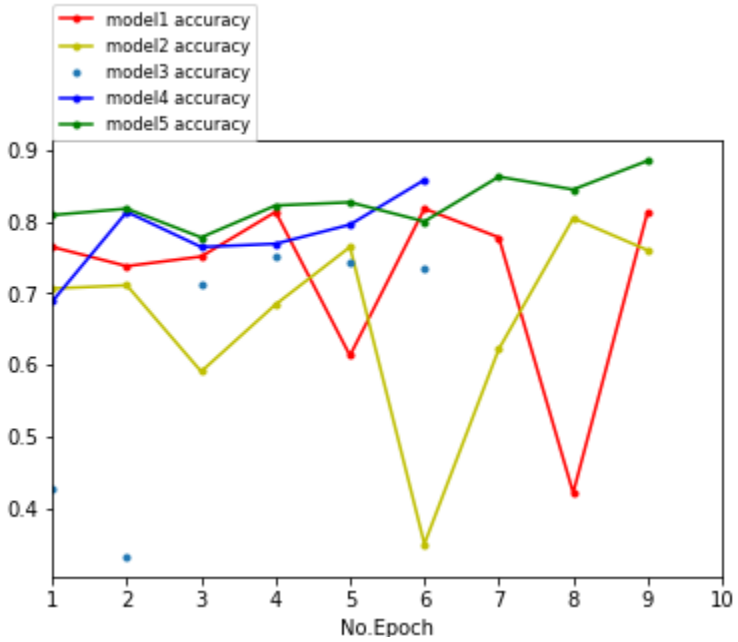
Since we achived good enhacenent in model number 4 we have decided to keep model 4 and change its optimizer to Adam.

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 125, 125, 250)     7000
_________________________________________________________________
activation_16 (Activation)   (None, 125, 125, 250)     0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 62, 62, 250)       562750
_________________________________________________________________
activation_17 (Activation)   (None, 62, 62, 250)       0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 31, 31, 250)       0
_________________________________________________________________
dropout_10 (Dropout)         (None, 31, 31, 250)       0
_________________________________________________________________
flatten_5 (Flatten)          (None, 240250)            0
_________________________________________________________________
dense_10 (Dense)             (None, 512)               123008512
_________________________________________________________________
activation_18 (Activation)   (None, 512)               0
_________________________________________________________________
dropout_11 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_11 (Dense)             (None, 4)                 2052
_________________________________________________________________
activation_19 (Activation)   (None, 4)                 0
=================================================================
Total params: 123,580,314
Trainable params: 123,580,314
Non-trainable params: 0
_________________________________________________________________
```

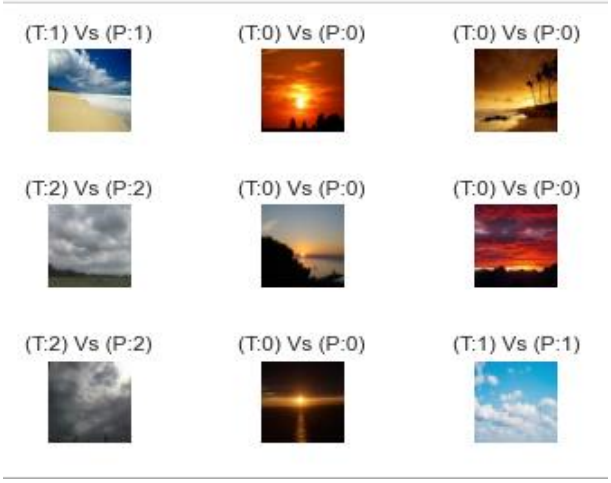
The best accuracy achieved by using validation dataset is 90.22% at epoch 10

Selected Model:

Model number 5 has been selected because its accuracy is the highest one, we reached to approximately 91% score by using this model and its execution time was in average, below figure illustrates the performance of model's vs number of epochs, Its Cleary noticeable that model 5 (green ones) is the best one …(See the below figure)



Below figure shows the accuracy of selected model:



T	True
P	Predicted

0	Sunrise
1	Shine
2	Cloudy
3	Rain

Key Findings:

1- Increases the number of layers impact badly the performance of model.
2- Running the models for high number of epochs does not leverage model performance, in contrast it impacts badly its performance.
3- Using leaky Relu activation function instead of Relu has impacted badly model performance.
4- Adam optimizer has shown significant distribution toward model performance.
5- Reducing Kernal size has shown significant distribution toward model performance.

Suggestions:

1- Reconducting this study by using more variety and size of images.
2- Using Function type model to evaluate its performance and weather significant achievement can be gained.
3- Using GPU instead of CPU to leverage machine hardware capabilities.

Hope you enjoy reading this short report and Happy learning my peer…