National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Indranil Bera
Student ID:	20144075
Programme:	MSc in Data Analytics **Year:** 2021-2022
Module:	MSc Research Project
Lecturer:	Vladimir Milosavljevic
Submission Due Date:	30/08/2022
Project Title:	Identify cloud patterns by both cloud images and meteorological parameters using hybrid deep learning model
Word Count:	1740 **Page Count:** 5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Indranil Bera

Date: 30/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Explanation of Questions

Indranil Bera
x20144075@student.ncirl.ie

Q1. What image manipulation and feature engineering steps could be applied to improve your models?

There are a few image manipulation and feature engineering methods which could be applied during model building. Below are those techniques which I found useful for this problem.

- **Geometric transformations -** The image dataset was small with 784 cloud images because the area considered for the study was a small area of Singapore. Geometric transformation is a popular technique to increase the number of images. Random rotation, shifting, and flipping methods could be applied to the image dataset to produce more images during the training of the model. These operations do not change the amount of information stored in the image. The easiest way to implement these methods is by using the Keras ImageDataGenerator library. The transformations can be only applied to training data of the cloud images, not to test or validation data.

- **Sharpening** – Sharp images can help algorithms to detect the edges accurately. Sharpness depends on the resolution and acutance. The resolution of the images here was fixed (125x125). Acutance measures the contrast of edges in an image. So if the sharpness of the images were increased then acutance was also increased. More contrast could be seen in the edges of the cloud images so it could be easier to detect edges and classify cloud types.

There are several methods related to image processing available to implement. My approach was to use fewer image processing techniques (i.e., Scaling and Resizing were used) but to rely on deep learning methods. So that the improvement of the hybrid model can be distinguished solely based on the algorithms applied.

Q2. During the data preparation step, the csv type data is processed through the weather API. What is the reason for that and what type of output you expect from that step?

The weather API was used to create the weather data for each image. The image dataset (SWIMCAT) had 5 folders for 5 cloud types. Each folder had one folder consisting of all the images of that type and a CSV file having data and time information of each image. Figure 1 illustrates the folder structure. From the CSV file date and time were taken as inputs to gather the weather parameters from the API.

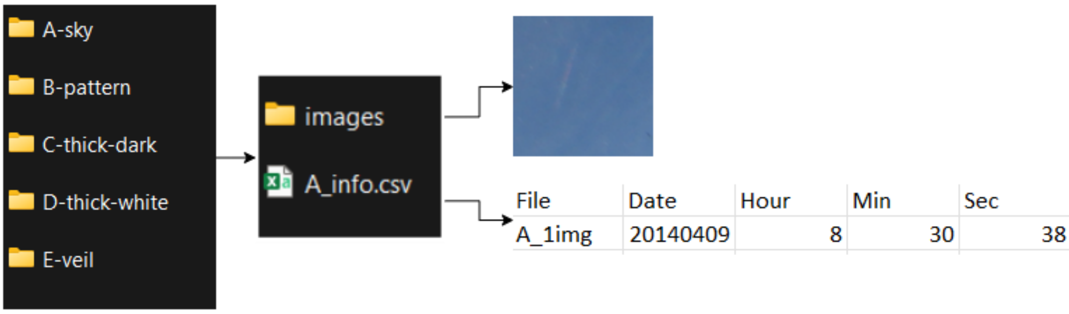

Figure 1 Folder structure of image dataset

The output of the API was comma-separated text values which were converted in a pandas data frame and stored as a CSV file after merging weather data of all cloud types. Figure 2 is the sample of the data frame.

	Temperature	Dew Point	Relative Humidity	Heat Index	Wind Speed	Wind Direction	Visibility	Cloud Cover
0	87.7	73.9	64.48	100.9	4.5	83.6	6.1	82.3

Figure 2 Sample Data frame

Q3. Model evaluation: It is unclear that which numerical number represent which cloud type. Can you clarify?

Apologies for not mentioning all the cloud types related to representing numbers used in the evaluation. Below are the cloud types -

0 – Clear sky
1 – Patterned clouds
2 – Thick dark clouds
3 – Veil clouds
4 – Thick white clouds

Q4. In your opinion, which features (image/ numerical) could contribute the most to the prediction of each cloud type?

Looking at the evaluation of this experiment, the MLP model with only numerical weather data had 84% accuracy. It had issues with overfitting. The F1-score was also low at 0.80 which indicated the detection of each cloud type was not that accurate. The VGG16 model with only image data had an accuracy of 88% and an F1-score of 0.83. This explains that image data had more information to contribute to the detection of cloud images than numerical weather data. The performance of the model was improved in the hybrid model with both image and numeric data. That means numeric data also helped to increase the accuracy. In my opinion image data had more contribution than numerical weather features in cloud-type classification.

Q5. Explain how transfer learning is performed in VGG-16 for your dataset.

Transfer learning is a technique in which a pre-trained model on huge data is used to train and predict on smaller data. Here in this project, VGG-16 was used to classify cloud images. The VGG-16 was pretrained in the 'ImageNet' dataset. ImageNet consists of 14 million images of 1000 categories which made this database popular for transfer learning method.

```
# Import the VGG16 model
from keras.applications.vgg16 import VGG16
model = VGG16(weights='imagenet')
```

Figure 3 VGG16 import

In the implementation of the code, at first VGG16 with 'imagenet' as weights was defined as shown in figure 3. So that model is loaded with the VGG16 pretrained with ImageNet dataset. Then at the end of the model definition, 'flatten' layer was added to convert the multi-dimension arrays from the pooling layers to a single-dimension vector. The last dense layer had 5 nodes as this classification had 5 types of clouds. Model compilation output in Figure 4 showed 616,293 trainable parameters were present in the model. This model then used to train on the training data and tested on the validation data. In this way transfer learning was implemented by VGG16 in this dataset.

```
flatten (Flatten)           (None, 800)           0

dense (Dense)               (None, 256)           205056

dense_1 (Dense)             (None, 64)            16448

dense_2 (Dense)             (None, 5)             325

=================================================================
Total params: 616,293
Trainable params: 616,293
Non-trainable params: 0
```

Figure 4 Model definition

Q6. In your opinion, how did concatenation of features help improve the accuracy of hybrid model? Is concatenation really addressing the multi-modality?

In the hybrid model, both images and numerical weather data were used. The extracted features from images and weather data were merged in the concatenation layer. The outputs of the CNN-based image classification network and MLP-based network acted as inputs for concatenate function. This concatenation operation increases the features space (collection of extracted features) by utilising features from both inputs of same size tensors. That's why the hybrid model was able to learn and detect new features better than other models. In this way, the concatenation operation improved the accuracy of the hybrid model.

In a multimodal network, multiple data types are combined to build an algorithm which can help to improve performance. In this experiment two types of data, numeric weather data and image data were used to build the model for classifying cloud types. The concatenation function of Keras functional API helped to merge the features from two neural networks. So to handle multi-modality it was necessary to merge the features which was seamlessly done by concatenation layer. That's why concatenation was important to address the multi-modality in this study.

Q7. Explain the evaluation performance metrics for class '1', '2' and '4'. What does F1 score of 1.00 mean? What insights you can gather from it? Do you think your model may be over-fitting here?

	precision	recall	f1-score	support
0	0.89	1.00	0.94	31
1	1.00	1.00	1.00	35
2	1.00	1.00	1.00	8
3	0.69	1.00	0.82	9
4	1.00	0.62	0.76	21
accuracy			0.92	104
macro avg	0.92	0.92	0.90	104
weighted avg	0.94	0.92	0.92	104

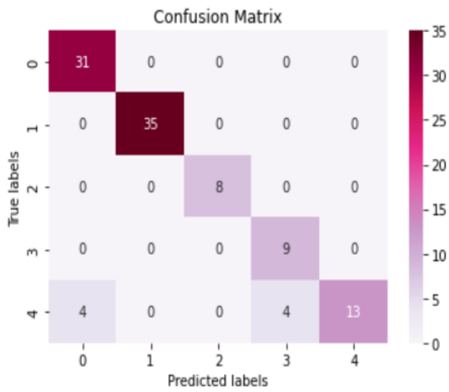

Figure 5 Performance metrices of Model 3

The classification report of Model 3 (Hybrid) in figure 5 shows class 1 and class 2 had precision, recall, and f1-score of 1.00. This means the model trained on train data was able to identify all the images of class 1 (Patterned cloud) and 2 (Thick dark clouds) accurately on the test data. 35 images of patterned clouds and 8 images of thick dark clouds were taken for testing and no single image was misclassified. In the case of class 4, the precision was 1 which means there was no false positive so all the thick white clouds were predicted correctly as thick white clouds. Recall and f1-score for this class were 0.62 and 0.76 respectively. This indicates thick white clouds were wrongly predicted into different categories which generated false negatives. The confusion matrix shows out of 21 images of thick white clouds 13 were classified correctly but 4 were misclassified into class 3 (Veil clouds) and the rest 4 were misclassified as class 0 (Clear sky).

F1-score is the harmonic mean of recall and precision. F1-score 1 denotes the value of recall and precision both are perfect means 1. So the model was able to classify all the images of this class correctly as discussed earlier for class 1 and class 2. F1-score is an indication of problems related to overfitting of a model but not always. Overfitting occurs when a designed algorithm performs well on training data but stops learning in the test set. In case of overfitting, the model minimizes training set error while increasing test set error. However, no such observations were noticed during the model's evaluation. Because the hybrid model had a larger feature space, it was able to classify images more accurately on test data with fewer images. Because of this, the f1-score had a value of 1. So I do not think there was any overfitting in the model.

Q8. How do you define the novelty of your research work?

After doing critical research of the papers in this domain, a few observations were made which concreated the goals and novelty of this project. Studies in the cloud classification domain were mostly done by images of clouds taken by radar, satellite systems etc. using CNN based algorithms. Very few studies were done using images captured from ground-based camera systems. The unavailability of such labelled datasets is the major reason for that. Multi-modal algorithms are getting popular in this domain and few researchers were found using both images and numeric weather data. Images used in the hybrid models were also not from ground-based sources. The weather parameters included for these kinds of studies were also limited to not more than two (i.e. humidity and temperature). So, there was a significant research gap in the classification of clouds captured from ground-based camera systems. Finally, 8 important numerical weather parameters were incorporated. So building a multi modal deep learning algorithm using ground-based cloud images and eight prominent numerical weather features make this research work unique and novel in the cloud classification domain.

Q9. Why did you not use VGG-16 for image feature extraction in hybrid models?

VGG-16 is one of the most popular architectures for CNN-based image classification. In the case of practical implementation of the developed hybrid algorithm image data needs to be processed at a faster rate from the captured device. Any IoT-enabled ground camera system or mobile camera can be integrated seamlessly for this purpose. So this hybrid model needed an architecture with faster processing with low latency. MobileNet uses depthwise separable layers and pointwise convolutional layers which reduced the no of parameters in the model. That's why MobileNet is a light weight architecture and consumes less computational resources than VGG-16. The MobileNet algorithm has low latency which makes it suitable for mobile or IoT related applications. Looking at the advantages of MobileNet architecture VGG-16 was not used for building the hybrid model.