Selected – 2 Project Cover Sheet

Faculty Name: *Faculty of Computers & Artificial Intelligence*

Course Name: *Selected Topics in Computer Science - 2*

Team Number: *30*

ID	Name
202000833	Mohamed Yousef Soliman
202000835	Mahmoud Ahmed Sayed Shoura
202000271	Hassan Mohammed Ahmed
202000910	Mostafa Mohammed Ramadan
202000359	Ziad Mohammed Arafa
202000124	Eslam Ayman Zaky Azab

Paper Details

Citation

Sharma, A., & Ismail, Z. S. (2022). Weather Classification Model Performance: Using CNN, Keras-Tensor Flow. In *ITM Web of Conferences* (Vol. 42). EDP Sciences.

Implemented Algorithms

Algorithm for Weather Classification Model

This algorithm accomplishes ten steps process to design and develop a weather classification model. These are the following steps:

[1]. Initiate a model

[2]. Import all of the required system defined libraries

[3]. Preparation of image dataset for training the machine and validating the machine model

[4]. Show graphically for classifying the training and validating data into distinct classes

[5]. Build a model using CNN

[6]. Compile a model

[7]. Train a model as per the training dataset

[8]. Validate a model as per the validation dataset

[9]. Display the performance of an implemented model graphically

[10]. Finish the process

Pseudo-code of the proposed algorithm

The pseudo-code of the proposed algorithm for a weather classification model, such as:

Step-1. *Initiate a model.*

Step-2. *To import all the required inbuilt libraries, such as : TensorFlow, Keras, NumPy, Matplotlib*

Step-3. *To prepare data for training the model and validating the model.*

- *Set path for image data collection*
- *Classify the image data into four classes: Cloud, Rain, Shine, and Sunrise.*
- *Divide image data into distinct batches, i.e., TRAINING_DIR and VALIDATION_DIR.*

Step-4. *Display bar-graphs*

- *Plot bar-graph for Classification of the training dataset.*
- *Plot bar-graph for Classification of the validating dataset.*

Step-5. *Build a new Weather Classification model by CNN.*

- *Design a sequential model with convolution 2D filters, kernel-size, activation function 'relu,' and padding function 'same'; maximum pooling size of 2D images; add a dense layer with activation function' softmax.'*
- *Summarize the model.*

Step-6. *Compile a new Weather Classification model.*

Step-7. *Train a new Weather Classification model as per the collected images dataset.*

Step-8. *Validate a new Weather Classification model as per the collected images dataset.*

Step-9. *Plot the line graphs to show the performance of 'Training and Validation Accuracy' and 'Training and Validation Loss.'*

Step-10. *End.*

Results

Analysis of the model's performance

Accuracy, validation accuracy, losses, and validation losses are approximately 94%, 92%, 18%, and 22%, respectively, as look in Figure-3 and Figure-4. It is pretty explained here is no overfitting in the line graphs. All four curves show similarities for training and validation of an implemented model.

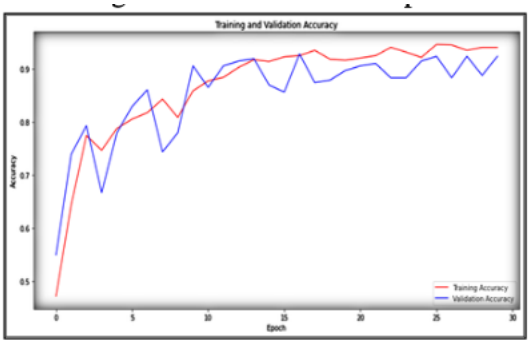


Fig. 3. Training and validation accuracy for the weather classification model.

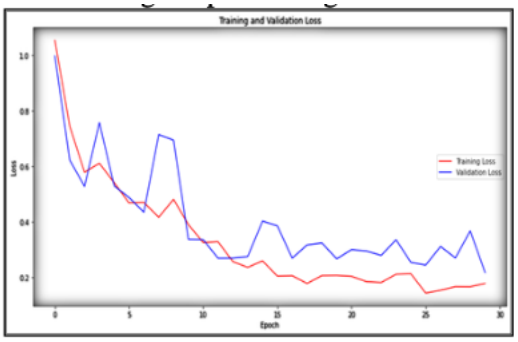


Fig. 4. Training and validation loss for the weather classification model.

Confusion matrix of the model's performance

This confusion matrix shows the results for total dataset images of 903 are [[57, 48, 73, 62], [37, 36, 42, 58], [54, 45, 47, 57], [51, 72, 66, 98]]), such as correct prediction for cloud, rain, shine, and sunrise are 57 out of 240, 36 out of 173, 47 out of 203 and finally 98 out of 287, respectively, look in Figure-5.

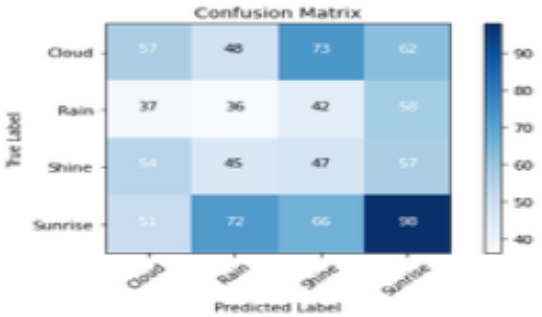


Fig. 5. Confusion matrix

Project Description Document

General Information on the selected dataset

The dataset used is called Weather Classification by Vijay Gupta (https://www.kaggle.com/datasets/vijaygiitk/multiclass-weather-dataset),

The dataset contains about *1500 labelled images* including the validation images. Images are *not of fixed dimensions* and the photos are of *different sizes*. Each image has only one weather category and are saved in separate folder as of the labelled class.

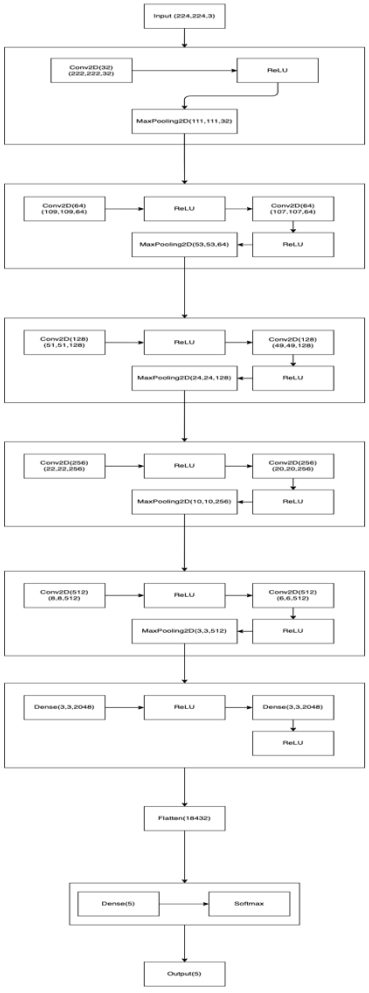
Each image have been rated for the weather condition on one of *5 classes*,
0 for *Cloudy*, 1 for *Foggy*, 2 for *Rainy*, 3 for *Shine* and 4 for *Sunrise*

Implementation Details

The ratio used for training – validation – testing is
80% (1198 images) - 10% (150 images) - 10% (150 images)
The images were reshaped to size (224,224) with 3 channels(red,green,blue), and the pixels were scaled over 0 and 1 by dividing them over 255. Also, Label encoding and Label Binarizing was used on the labels

We first implemented the model using the paper's method,
Using 9 ***Conv2D*** layers, 5 ***MaxPooling2D*** layers, 3 ***Dense*** layers and 1 ***Flatten*** layer.In a total of 18 trainable layers, and all of the

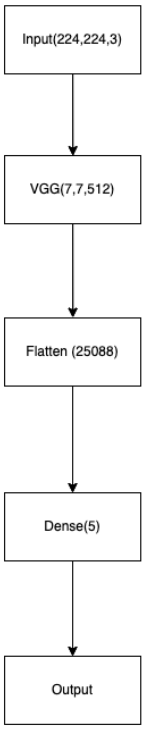
Here is the block diagram:



But also, we modified the paper's method to yield better results by using Transfer Learning, so our ***modified model*** consists of 3 layers, **VGG19(non-trainable**), **Flatten**, and **Dense**.

VGG19 consists of 19 layers that are pre-trained on millions of different images from the ImageNet database, it helps in saving training time, and better performance of neural networks and not needing a lot of data.

Here is the block diagram:



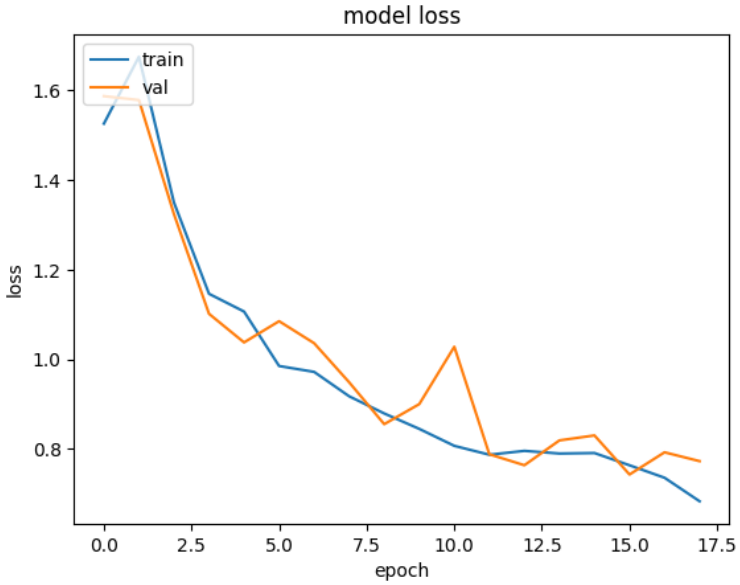
Hyperparameters used in both models:

- Optimizer: adam
- Loss: categorical crossentropy
- Batch size: 32
- Epochs: 30

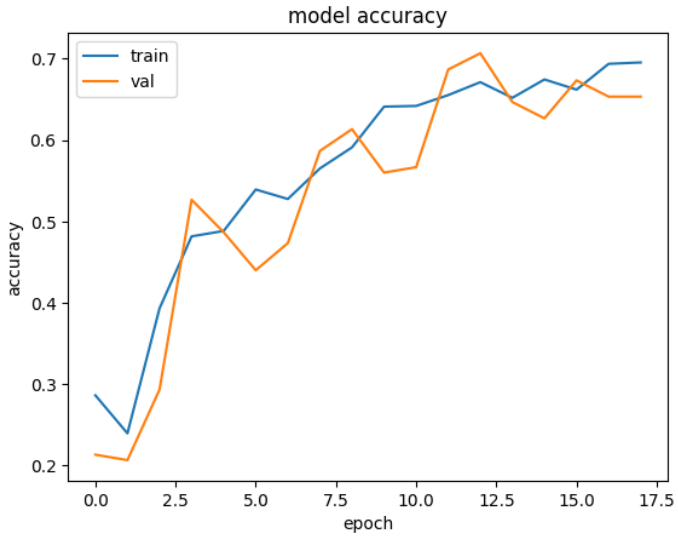
Results Details

-Paper's implementation:

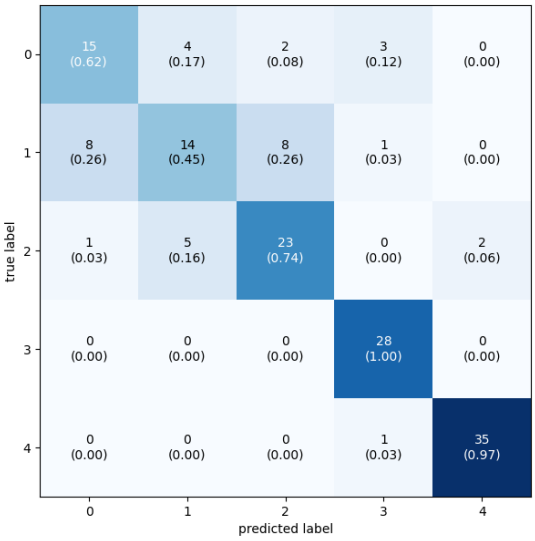
- Loss curve:



- Accuracy curve:



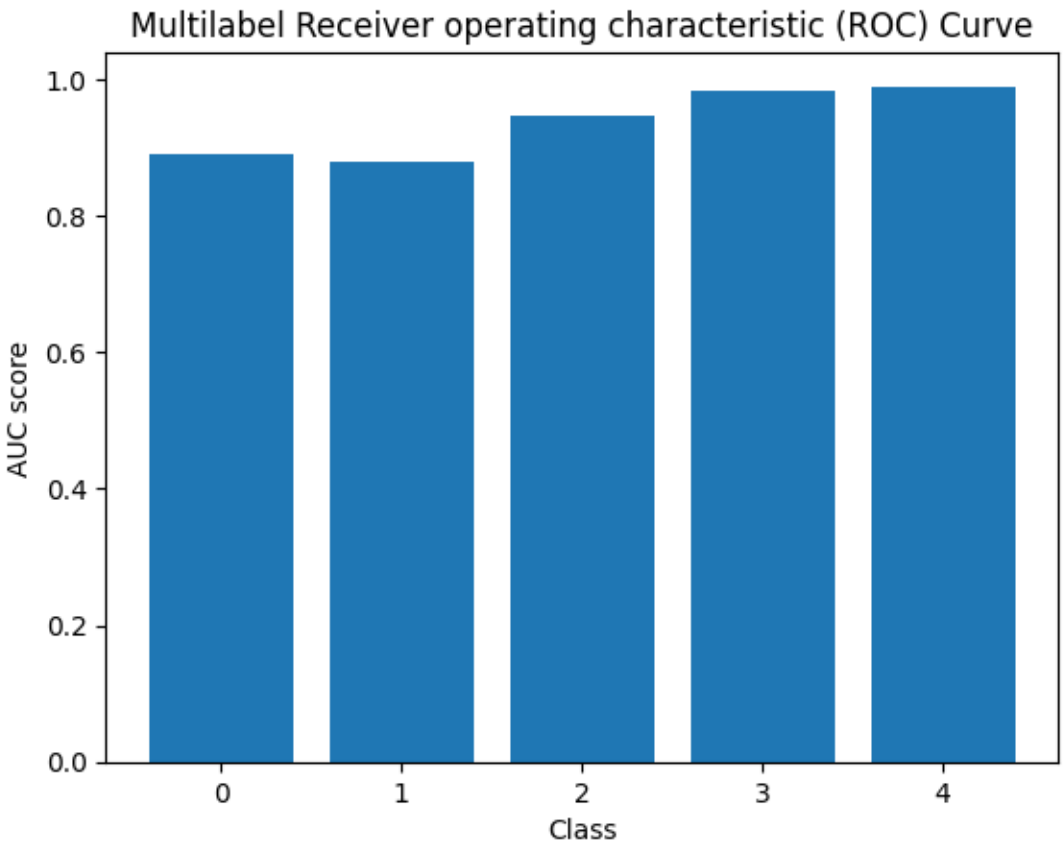
- Confusion Matrix:



- Accuracy, Precision, Recall, F1:

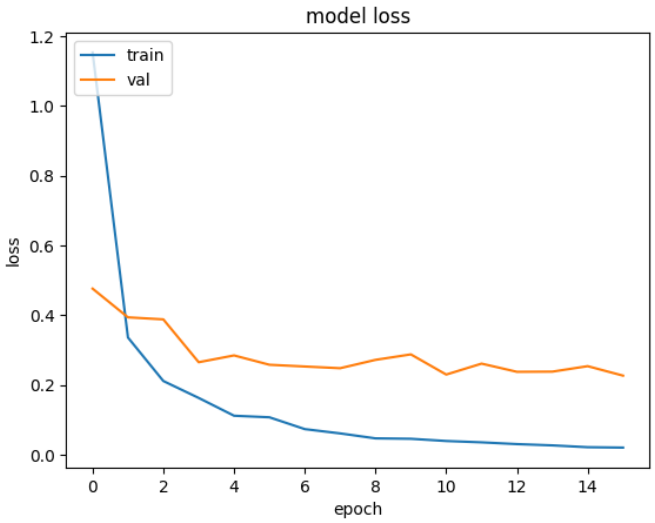
	precision	recall	f1-score	support
0	0.62	0.62	0.62	24
1	0.61	0.45	0.52	31
2	0.70	0.74	0.72	31
3	0.85	1.00	0.92	28
4	0.95	0.97	0.96	36
accuracy			0.77	150
macro avg	0.75	0.76	0.75	150
weighted avg	0.76	0.77	0.76	150

- ROC:

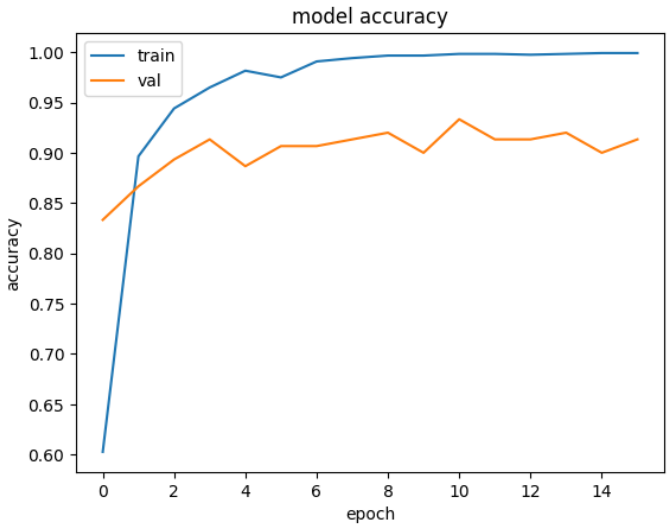

Multilabel Receiver operating characteristic (ROC) Curve
AUC score
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
Class

-Modified implementation:

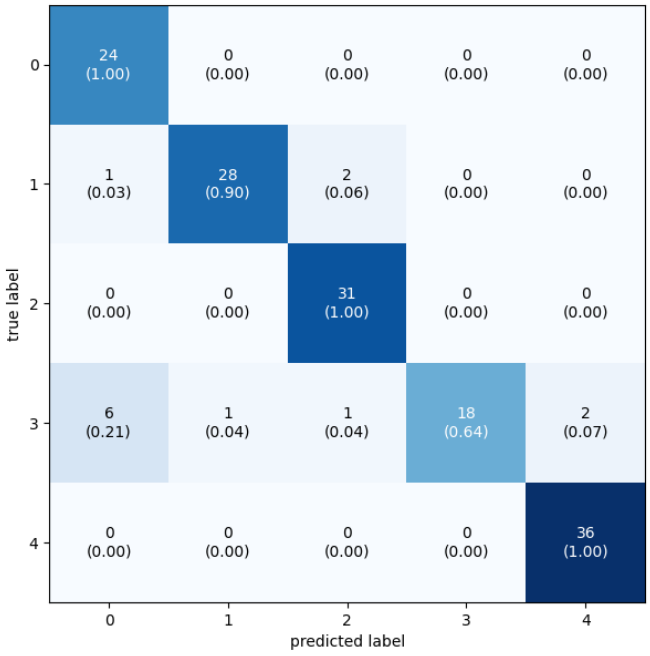
- Loss curve:



- Accuracy curve:



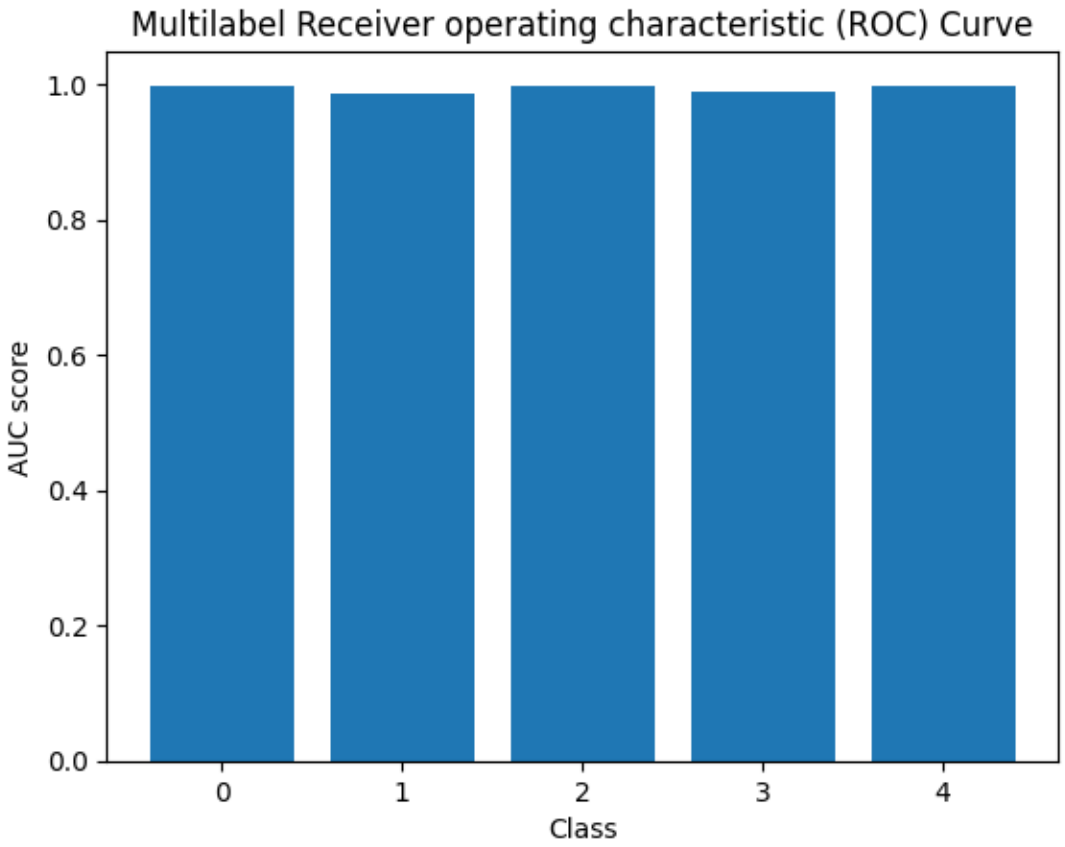
- Confusion Matrix:



- Accuracy, Precision, Recall, F1:

	precision	recall	f1-score	support
0	0.77	1.00	0.87	24
1	0.97	0.90	0.93	31
2	0.91	1.00	0.95	31
3	1.00	0.64	0.78	28
4	0.95	1.00	0.97	36
accuracy			0.91	150
macro avg	0.92	0.91	0.90	150
weighted avg	0.93	0.91	0.91	150

- ROC:


Multilabel Receiver operating characteristic (ROC) Curve
AUC score
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
Class