King Fahd University of Petroleum and Minerals

Cloud Classification for Rain Precipitation.

ICS 471: Artificial Neural Networks and Deep Learning

12/19/2020

For

Dr. Aiman H. El-Maleh

Group members:

Ahmad Alsaqqa	*201541790*
Mohammed Alghamdi	*201679060*
Mohammad Alsulimani	*201685360*

Contents

Team Work Distribution

Mohammad Alghamdi discovered a very useful dataset which contains around 2.5k images of clouds, split into 11 classes. To ensure consistency in implementation, the team developed the notebook used for training together. The notebook then was distributed to the rest of the team and we began training our neural network using different models. Each team member carried out multiple training routines to find the optimal solution. The models that were used in our experiment include, but not limited to, Resnet101, MobileNet and GoogleNet. Each team member has chosen different models to train. The results, then where compared and the final model was chosen. We have managed to run more than 25 training routines. Our comparison for the training consisted of the following performance measures:

1. Training accuracy
2. Validation Accuracy
3. Training loss
4. Validation loss

Finally, InceptionV3_v6, was chosen to be implemented in an actual application because it scored the highest accuracy among the other models. Our application is a Telegram bot that accepts images from clients, passes them through our classifier, and finally returns the prediction it has given out of the 11 different cloud classes.

1. Introduction.

The impact of deep learning applications has been dramatic ever since the early 2000s. Thanks to the rapid improvement of deep learning techniques, such as the convolutional neural networks, development of technologies that solve a wider range of real-world problems has become widely accessible. Our team found interest in studying the clouds and how they differ.

Our aim in this project is to utilize deep learning techniques to develop a system that classifies clouds. A picture of a cloud shall be inputted to the system and, using a trained neural network model, a classification of the input picture is outputted. We will be utilizing a variation of CNNs that is designed and curated to classify clouds.

2. Inspiration.

Clouds are unique and interesting natural phenomena. Cloud shapes, height, and velocity are astonishing to observe and think about the greatness of the creator. Clouds can be a sign for a nice calm day or maybe a sign to go back home before the rain started. Amazingly old farmers and people how are affected by rain can determine a rainy cloud from others. Sheikh Saeed Al Kamali in the YouTube video tells how Arabs were capable of forecasting rain from only clouds (Al Kamali, 2018). As a result, using deep learning to predict rain from a picture taken by a mobile phone will be amazing.

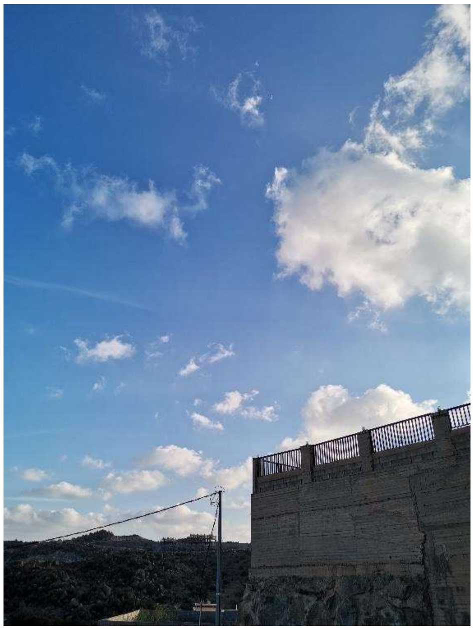
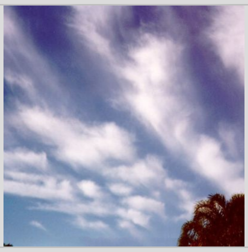
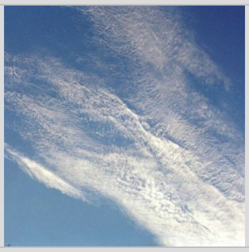
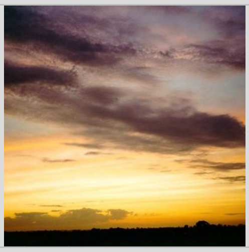
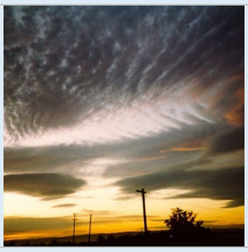
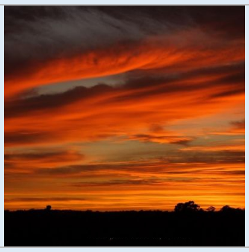
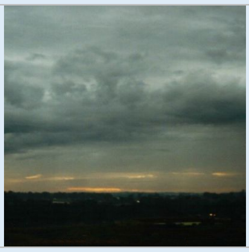


Figure01: Picture taken Saturday 12/19/2020 at 4:36 PM.

3. Problem Description.

Determining different types of clouds is challenging and requires high experience. Clouds were classified into seven classes based on their shape and altitude in December 1802 by Luke Howard (Pedgley, 2003). However, different and various classifications came after Luke. The problem goal is to classify ten different cloud types using deep learning techniques. Moreover, we can classify clouds more into rain clouds or non-rain clouds according to the Bureau of Meteorology Australia's official weather forecasts (Cloud Types and Precipitation, n.d.). The ten different cloud types can be shown in table01. In summary, the team is planning to develop a Telegram bot that can classify a cloud from an image taken by a mobile phone and give rainfall precipitation forecasting.

High Altitude (above 6 km)		
Cirrus	Cirrocumulus	Cirrostratus
		
No Precipitation	No Precipitation	No Precipitation
Middle Altitude (2.5 to 6 km)		
Altocumulus	Altostratus	Nimbostratus
		
May produce light showers.	Rain or snow.	Heavier intensity rain or snow.

Low Altitude (below 2.5 km)		
Stratocumulus	Stratus	Cumulus
Drizzle.	Drizzle.	Showers or snow.
Cumulonimbus.		
Showers or snow.		
Generated by Airplanes (contrail)		
No precipitation.		

Table01: Clouds Types and Precipitation.

4. Dataset.

The dataset is known as Cirrus Cumulus Stratus Nimbus (CCSN). The dataset consists of 11 different cloud categories. CCSN includes an interesting type of clouds that are generated by human activity which is known as a contrail. Contrail is a type of cloud generated by airplanes. CCSN was used to develop CloudNet which is a neural network consisting of multiple convolution layers and max-pooling layers that can classify cloud using ground imagery (Zhang, 2018). The dataset contains 2543 cloud images. Table02 shows the distribution of the images for the different classes. Some preprocessing techniques will be used to increase the dataset size. Rotation and adding noise are techniques that can be used to increase the number of images.

Name	Number of Instances
Cirrus	139

Cirrocumulus	268
Cirrostratus	287
Altocumulus	221
Altostratus	188
Nimbostratus	274
Stratocumulus	340
Stratus	202
Cumulus	182
Cumulonimbus	242
Contrail	200

Table02: Image Distribution based on Classes.

5. Deep Learning Network Type.

Understanding the nature of our problem will help in determining the best deep learning network. The project goal is to classify ground cloud images into 11 different classes. As a result, the problem is an image classification problem. So from our previous knowledge in studying various deep learning networks; The team decided to use Convolution Neural Network (CNN). CNN has been used enormously in classification problems like face classification, face mask detection, cancer image classification, and many applications in the field of deep learning.

6. Proposed Structures.

Deciding the optimal architecture for a problem is an art, not a science. However, there are some known facts for a better neural network like a deeper network (more layers), and differentiable activation functions (like sigmoid or ReLU). However, PyTorch provides efficient pre-made neural network architectures that can be used immediately. The pre-made architectures have been used in different classification problems and have achieved astonishing results. As a result, different CNN architectures have been used which are:

1. Resnet101
2. Resnet152
3. MobileNetV2
4. VGG13
5. Googlenet
6. InceptionV3

The team set the published architecture (CloudNet) in (Zhang, 2018) as a reference point. As shown in figure02, CloudNet architecture is much smaller than the different architectures that have been used by the team. As a result, the team was more confident that the assigned architectures will achieve better results than CloudNet. All the architectures were modified by adding Center-Loss layer, and the output layer at the end of the network. Also, Cross-Entropy criterion was used to apply Softmax function to the last layer.

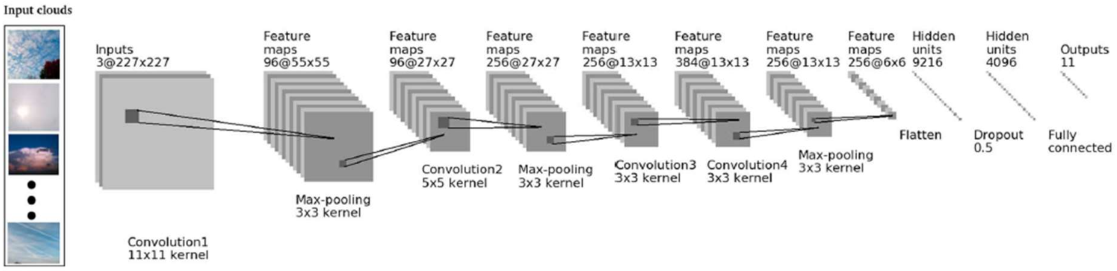
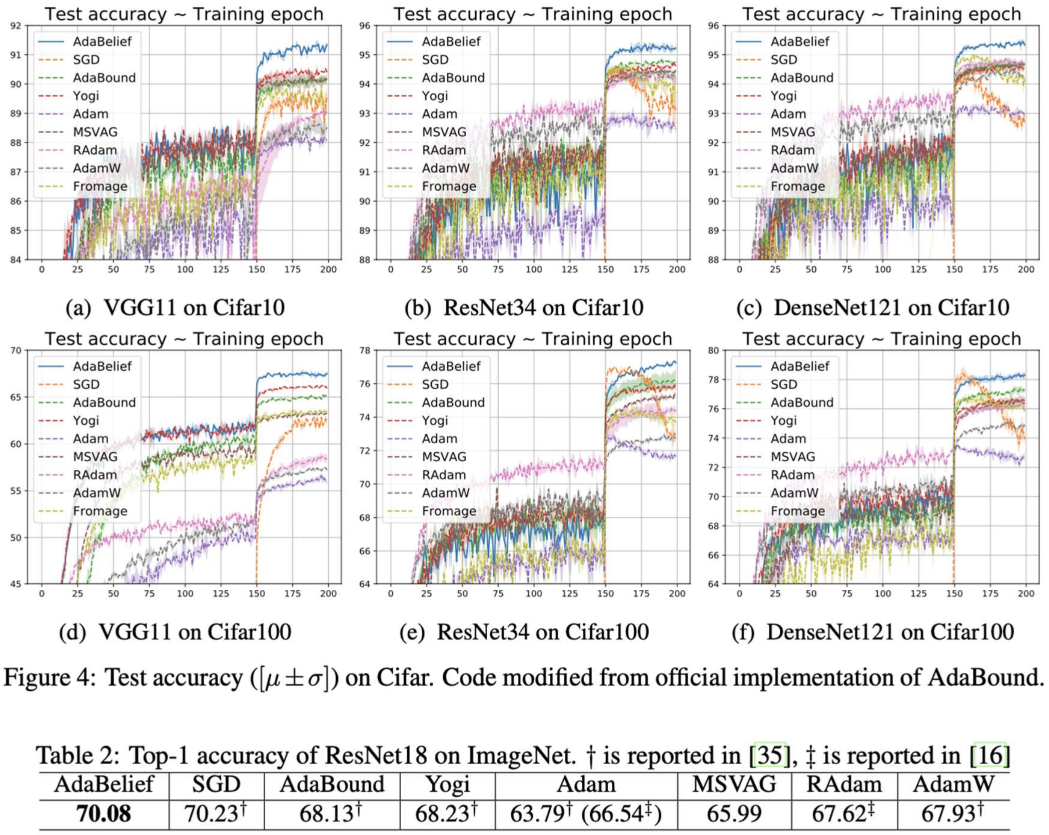


Figure02: CloudNet Architecture

7. Optimization and Data Augmentation Techniques.

Previous assignments and gained knowledge throughout the deep learning course helped the team in deciding the best optimization techniques and data augmentation. The team used AdaBelief optimizer because it achieved better results than other known optimizers like Stochastic Gradient Descent (SGD) and Adam as shown in figure03 according to (Zhuang, 2020). Moreover, SGD was used with the Center-Loss criterion to assist the classification results.



(a) VGG11 on Cifar10 (b) ResNet34 on Cifar10 (c) DenseNet121 on Cifar10

(d) VGG11 on Cifar100 (e) ResNet34 on Cifar100 (f) DenseNet121 on Cifar100

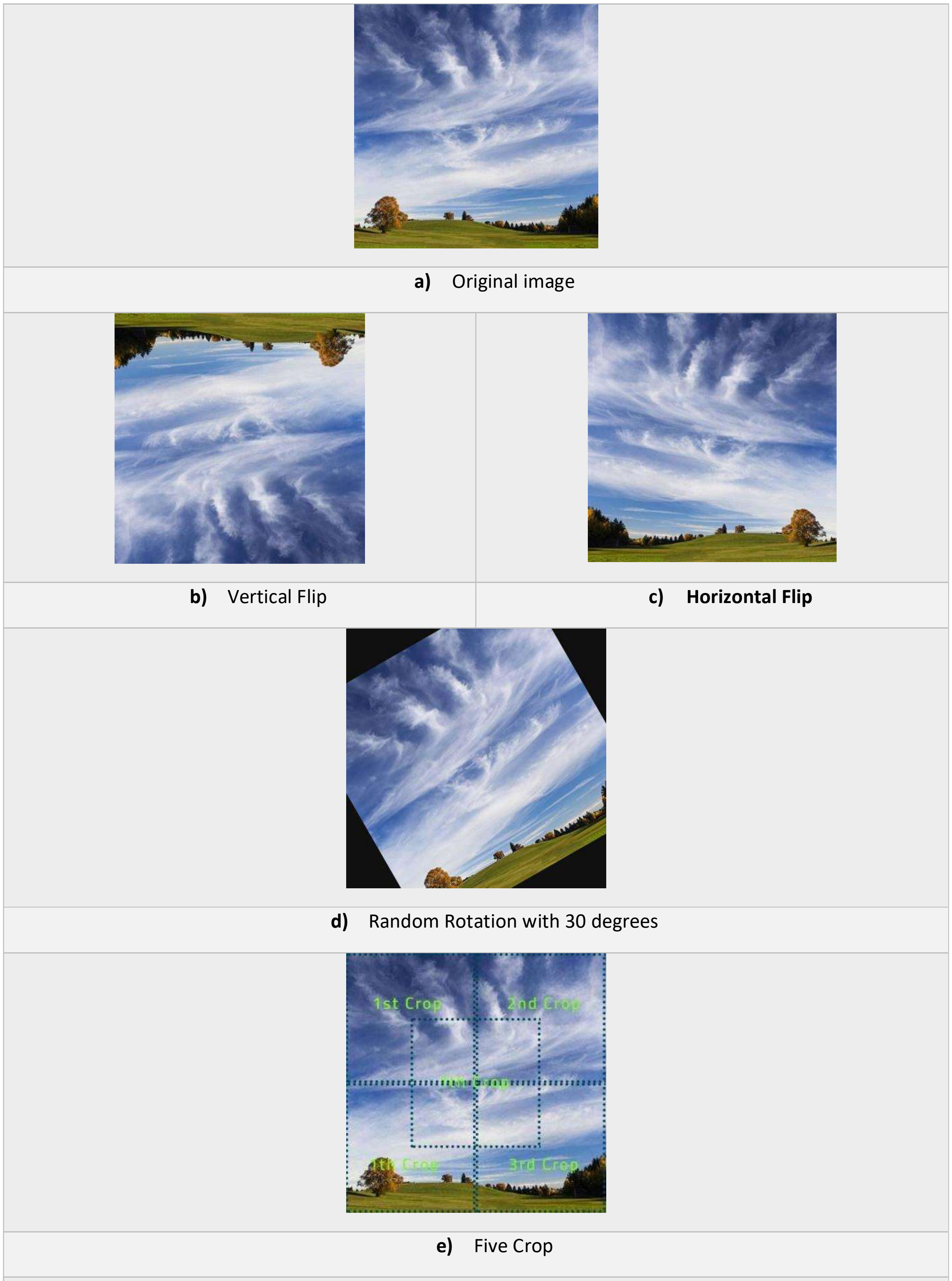
Figure 4: Test accuracy ($[\mu \pm \sigma]$) on Cifar. Code modified from official implementation of AdaBound.

Table 2: Top-1 accuracy of ResNet18 on ImageNet. † is reported in [35], ‡ is reported in [16]

AdaBelief	SGD	AdaBound	Yogi	Adam	MSVAG	RAdam	AdamW
70.08	70.23†	68.13†	68.23†	63.79† (66.54‡)	65.99	67.62‡	67.93†

Figure03: AdaBelief Performance with other Optimizers.

As discussed before, some techniques will be applied to the dataset to increase the number of images. Increasing the number of images will help the model to generalize and reduce overfitting. Pytorch provides some data augmentation techniques under torchvision transforms model. The suggested transformers are five crops, random horizontal rotation, random vertical rotation, and random rotation. Before applying the suggested transformers, the team did some testing and experiments to decide on the best transformers. As figure04 shows the different transformers, the random vertical rotation was excluded because the ground appears as the sky in some images. The random rotation was accepted with a small degree of rotations less than 20 degrees because more will result in large black corners. Using Fivecrop helped increase the size of the dataset by 5, so the total number of images increased from around 2.5k to around 12.5k.



a) Original image

b) Vertical Flip

c) **Horizontal Flip**

d) Random Rotation with 30 degrees

e) Five Crop

Figure04: Different Data Augmentation.

8. Experimental results.

20 models of different architectures and variables were trained by the team to decide and find the optimal model. The optimal model as discussed before will be used in the telegram bot. The model accuracies for training and validation are shown in figure05. **InceptionV3_v6** achieved the best results and will be used in the telegram bot. For more information about the used models review the Appendix.

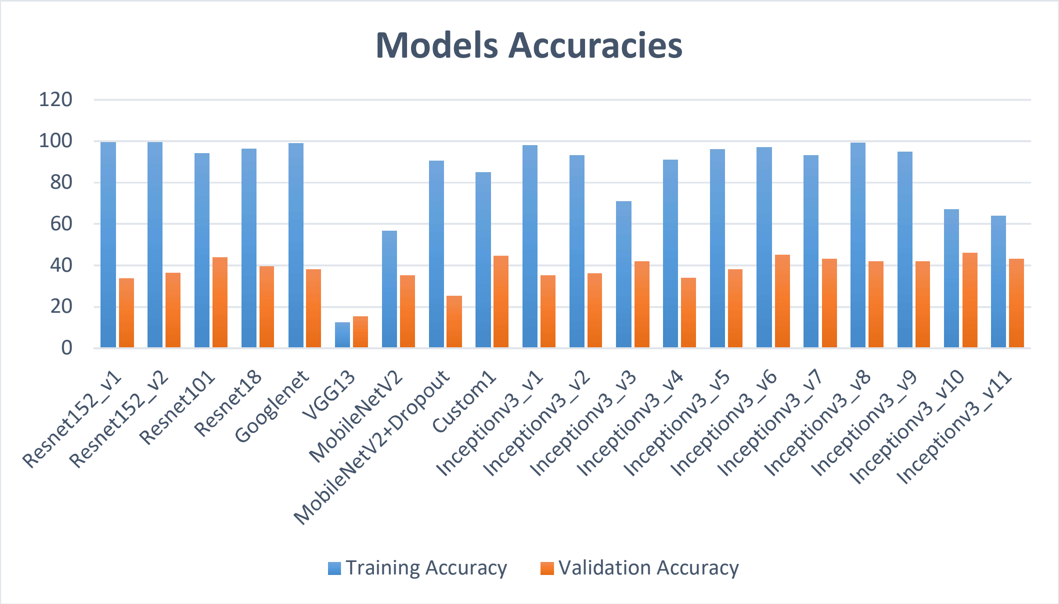


Figure05: Models Accuracies.

9. Existing Solution.

The team has used the same dataset that has been used by the existing solution (CloudNet). CloudNet claims that they achieved a near-perfect classification. However, our team doubts their claim based on our experiments and observations throughout the development of this project. The solution didn't provide the training accuracy, nor testing and only provided results for what they call (experimental classification results) as shown in figure07. The average accuracy of the CloudNet is 87.72%. So if we considered the average accuracy as the training accuracy our team achieved a state-of-art solution. On the other hand, if the average accuracy is considered as the testing (validation) accuracy CloudNet is a better model.

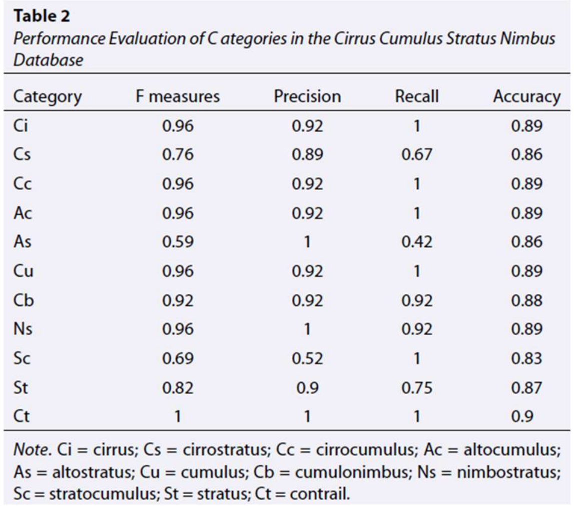


Table 2
Performance Evaluation of C ategories in the Cirrus Cumulus Stratus Nimbus Database

Category	F measures	Precision	Recall	Accuracy
Ci	0.96	0.92	1	0.89
Cs	0.76	0.89	0.67	0.86
Cc	0.96	0.92	1	0.89
Ac	0.96	0.92	1	0.89
As	0.59	1	0.42	0.86
Cu	0.96	0.92	1	0.89
Cb	0.92	0.92	0.92	0.88
Ns	0.96	1	0.92	0.89
Sc	0.69	0.52	1	0.83
St	0.82	0.9	0.75	0.87
Ct	1	1	1	0.9

Note. Ci = cirrus; Cs = cirrostratus; Cc = cirrocumulus; Ac = altocumulus; As = altostratus; Cu = cumulus; Cb = cumulonimbus; Ns = nimbostratus; Sc = stratocumulus; St = stratus; Ct = contrail.

Figure06: CloudNet Results.

10. Best Model Deployment.

As discussed before, InceptionV3_v6 achieved the best accuracies and will be deployed into production using the Telegram Bot. Telegram provides an easy to use API that can be used to develop a simple chatbot. The bot will receive a cloud image from a user. Then the bot will send the prediction back to the user. The bot will be deployed on Amazon Web Services (AWS) EC2 instance which is a virtual machine that runs in the cloud. The bot's name is @TheMagnificentOneBot. The bot can be found using the Telegram app either in IOS or Android.

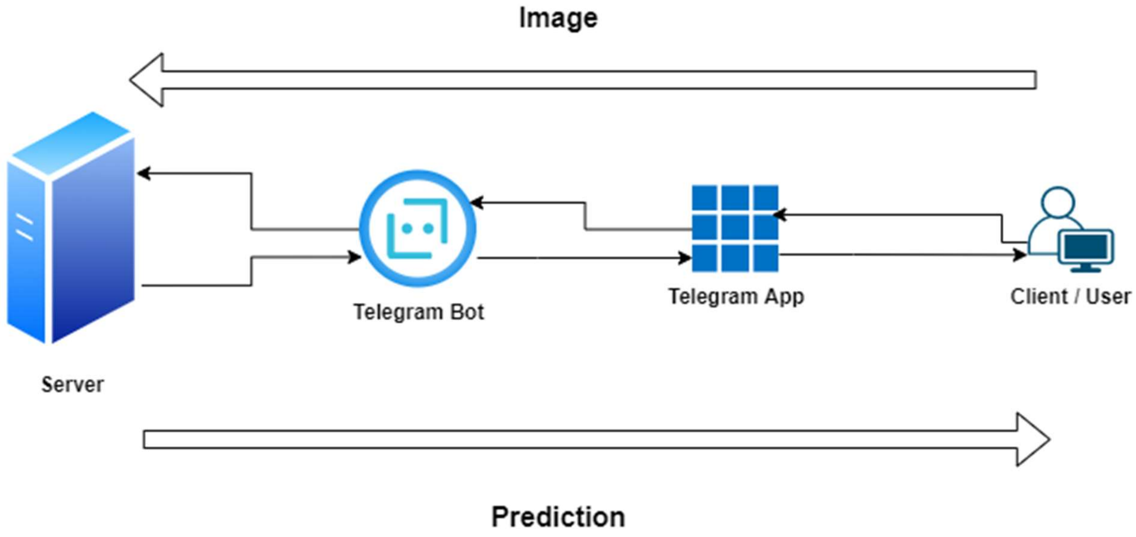


Figure07: Bot Architecture.

11. Suggestions.

The team is ambitious and would like to complete searching and developing more accurate models. However, throughout our search and experiment, the team felt the lack of datasets regarding ground-based cloud imagery. Some of the best papers are done by the Chinese and use private datasets. A small suggestion to KFUPM and to the College of Computer Science and Engineering is to build a system that collects weather data like temperature, humidity, wind speed, etc… and cloud images that are classified into cloud type and rain precipitation. Moreover, cooperation with different universities around the kingdom to host the system to collect datasets that can be used by Saudis and relate to Saudis.

12. Appendix.

Model Training Template	
Network Name	Resnet152_v1
Batch Size	10
New Image Size	150x150
Used Transforms	● Random horizontal flip ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    lr=1e-2,`
	`    eps=1e-16,`
	`    betas=(0.9,0.999),`
	`    weight_decouple=True,`
	`    weight_decay=5e-5,`
	`    rectify = False,`
	`    amsgrad=False`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    lr: 0.5`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.9938
Train Loss	0.0115
Validation Accuracy	0.3381
Validation Loss	5.4378
Epoch	`905`
Total Training Time	`21.12 hours`
Finish Date	5/12/2020

Model Training Template	
Network Name	GoogleNet
Batch Size	10
New Image Size	150x150
Used Transforms	● Random horizontal flip ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    lr=1e-2,`
	`    eps=1e-16,`
	`    betas=(0.9,0.999),`
	`    weight_decouple=True,`
	`    weight_decay=5e-5,`
	`    rectify = False,`
	`    amsgrad=False`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    lr: 0.5`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.9893
Train Loss	0.0232
Validation Accuracy	0.3814
Validation Loss	4.683
Epoch	`1000`
Total Training Time	`11.67 hours`
Finish Date	8/12/2020

Model Training Template	
Network Name	Resnet152_v2

Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Resize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    lr=1e-2,`
	`    eps=1e-16,`
	`    betas=(0.9,0.999),`
	`    weight_decouple=True,`
	`    weight_decay=5e-5,`
	`    rectify = False,`
	`    amsgrad=False`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    lr: 0.5`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.9948
Train Loss	0.0096
Validation Accuracy	0.3647
Validation Loss	5.033
Epoch	`554`
Total Training Time	`15.32 hours`
Finish Date	10/12/2020

Model Training Template	
Network Name	Resnet101
Batch Size	5
New Image Size	400x400

Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.9410
Train Loss	0.5517
Validation Accuracy	0.4391
Validation Loss	3.6819
Epoch	`468`
Total Training Time	`22:33:31.677453`
Finish Date	5/12/2020

Model Training Template	
Network Name	Resnet18
Batch Size	20
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.9629
Train Loss	0.0738
Validation Accuracy	0.3958
Validation Loss	3.8842
Epoch	`350`
Total Training Time	`3:00:37.060750s`
Finish Date	14/12/2020

Model Training Template	
Network Name	VGG13
Batch Size	20
New Image Size	400
Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	12.47%
Train Loss	2.3746
Validation Accuracy	15.47%
Validation Loss	2.3681
Epoch	100
Total Training Time	00:51:59
Finish Date	12/12/2020

Model Training Template	
Network Name	MobileNetV2
Batch Size	5
New Image Size	400
Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	56.57%
Train Loss	1.2783
Validation Accuracy	35.26%
Validation Loss	1.9870
Epoch	950
Total Training Time	8:58:52
Finish Date	13/12/2020

Model Training Template	
Network Name	MobileNetV2 + dropout
Batch Size	10
New Image Size	200
Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees ● Five-crop
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	90.51%
Train Loss	0.2947
Validation Accuracy	25.29%
Validation Loss	3.1161
Epoch	1000
Total Training Time	7:43:06
Finish Date	13/12/2020

Model Training Template	
Network Name	Custom1 [3,64,64,32]: ● Conv2d ○ Kernal: 7x7 ○ Stride: 2x2 ○ Padding: 3x3 ○ Bias: False ● Batchnorm2d ● ReLU ● MaxPool
Batch Size	20
New Image Size	400
Used Transforms	● Random horizontal flip ● Random rotation of maximum 20 degrees
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`amsgrad: False`
	`betas: (0.9, 0.999)`
	`eps: 1e-16`
	`lr: 0.01`
	`weight_decay: 5e-05`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`dampening: 0`
	`lr: 0.5`
	`momentum: 0`
	`nesterov: False`
	`weight_decay: 0`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.8506
Train Loss	0.4716
Validation Accuracy	0.4443
Validation Loss	2.1925
Epoch	308
Total Training Time	2:11:37.560854
Finish Date	17/12/2020

Model Training Template	
Network Name	Inceptionv3_v1
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize

Initialize the parameters	xavier_normal
Optimizer Label Loss	AdaBelief (Parameter Group 0 amsgrad: False betas: (0.9, 0.999) eps: 1e-16 lr: 0.01 weight_decay: 5e-05)
Optimizer Center Loss	Optimizer Center Loss: SGD (Parameter Group 0 dampening: 0 lr: 0.5 momentum: 0 nesterov: False weight_decay: 0)
Criterion Label Loss	CrossEntropyLoss()
Criterion Center Loss	CenterLoss()
feat_dim	10
Scheduler	None
Train Accuracy	0.98
Train Loss	0.06
Validation Accuracy	0.35
Validation Loss	3.95
Epoch	240
Total Training Time	3.53 H
Finish Date	1/12/2020

Model Training Template	
Network Name	Inceptionv3_v2 (-3 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	AdaBelief (
	Parameter Group 0
	    amsgrad: False
	    betas: (0.9, 0.999)
	    eps: 1e-16
	    lr: 0.01
	    weight_decay: 5e-05
	)

Optimizer Center Loss	Optimizer Center Loss: SGD (
	Parameter Group 0
	    dampening: 0
	    lr: 0.5
	    momentum: 0
	    nesterov: False
	    weight_decay: 0
	)

Criterion Label Loss	CrossEntropyLoss()
Criterion Center Loss	CenterLoss()
feat_dim	10
Scheduler	None
Train Accuracy	0.93
Train Loss	0..23
Validation Accuracy	0.36
Validation Loss	3.17
Epoch	135
Total Training Time	3.78 H
Finish Date	2/12/2020

Model Training Template	
Network Name	Inceptionv3_v3 (-5 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.71
Train Loss	0..86
Validation Accuracy	0.42
Validation Loss	1.91
Epoch	`100`
Total Training Time	`2.53 H`
Finish Date	3/12/2020

Model Training Template	
Network Name	Inceptionv3_v4 (-7 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.91
Train Loss	0.32
Validation Accuracy	0.34
Validation Loss	2.54
Epoch	`120`
Total Training Time	`1.42 H`
Finish Date	3/12/2020

Model Training Template	
Network Name	Inceptionv3_v5 (-9 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal
Optimizer Label Loss	AdaBelief (Parameter Group 0 amsgrad: False betas: (0.9, 0.999) eps: 1e-16 lr: 0.01 weight_decay: 5e-05)
Optimizer Center Loss	Optimizer Center Loss: SGD (Parameter Group 0 dampening: 0 lr: 0.5 momentum: 0 nesterov: False weight_decay: 0)
Criterion Label Loss	CrossEntropyLoss()
Criterion Center Loss	CenterLoss()
feat_dim	10
Scheduler	None
Train Accuracy	0.96
Train Loss	0.11
Validation Accuracy	0.38
Validation Loss	3.19
Epoch	220
Total Training Time	2.12 H
Finish Date	4/12/2020

Model Training Template	
Network Name	Inceptionv3_v6 (-10 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.97
Train Loss	0.09
Validation Accuracy	0.45
Validation Loss	3.29
Epoch	`340`
Total Training Time	`2.89 H`
Finish Date	5/12/2020

Model Training Template	
Network Name	Inceptionv3_v7 (-11 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.93
Train Loss	0.24
Validation Accuracy	0.43
Validation Loss	2.5
Epoch	`200`
Total Training Time	`1.56 H`
Finish Date	5/12/2020

Model Training Template	
Network Name	Inceptionv3_v8 (-12 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal
Optimizer Label Loss	AdaBelief (Parameter Group 0 amsgrad: False betas: (0.9, 0.999) eps: 1e-16 lr: 0.01 weight_decay: 5e-05)
Optimizer Center Loss	Optimizer Center Loss: SGD (Parameter Group 0 dampening: 0 lr: 0.5 momentum: 0 nesterov: False weight_decay: 0)
Criterion Label Loss	CrossEntropyLoss()
Criterion Center Loss	CenterLoss()
feat_dim	10
Scheduler	None
Train Accuracy	0.99
Train Loss	0.04
Validation Accuracy	0.42
Validation Loss	5.09
Epoch	940
Total Training Time	7.36 H
Finish Date	6/12/2020

Model Training Template	
Network Name	Inceptionv3_v9 (-13 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.95
Train Loss	0.17
Validation Accuracy	0.42
Validation Loss	3.18
Epoch	480
Total Training Time	3.6 H
Finish Date	12/12/2020

Model Training Template	
Network Name	Inceptionv3_v10 (-14 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	`AdaBelief (`
	`Parameter Group 0`
	`    amsgrad: False`
	`    betas: (0.9, 0.999)`
	`    eps: 1e-16`
	`    lr: 0.01`
	`    weight_decay: 5e-05`
	`)`

Optimizer Center Loss	`Optimizer Center Loss:  SGD (`
	`Parameter Group 0`
	`    dampening: 0`
	`    lr: 0.5`
	`    momentum: 0`
	`    nesterov: False`
	`    weight_decay: 0`
	`)`

Criterion Label Loss	`CrossEntropyLoss()`
Criterion Center Loss	`CenterLoss()`
feat_dim	10
Scheduler	None
Train Accuracy	0.67
Train Loss	0.99
Validation Accuracy	0.46
Validation Loss	1.71
Epoch	380
Total Training Time	3.1 H
Finish Date	17/12/2020

Model Training Template	
Network Name	Inceptionv3_v11 (-15 layer)
Batch Size	10
New Image Size	400x400
Used Transforms	● Random horizontal flip ● Five-crop ● Normalize
Initialize the parameters	xavier_normal

Optimizer Label Loss	AdaBelief (
	Parameter Group 0
	    amsgrad: False
	    betas: (0.9, 0.999)
	    eps: 1e-16
	    lr: 0.01
	    weight_decay: 5e-05
	)

Optimizer Center Loss	Optimizer Center Loss: SGD (
	Parameter Group 0
	    dampening: 0
	    lr: 0.5
	    momentum: 0
	    nesterov: False
	    weight_decay: 0
	)

Criterion Label Loss	CrossEntropyLoss()
Criterion Center Loss	CenterLoss()
feat_dim	10
Scheduler	None
Train Accuracy	0.64
Train Loss	1.26
Validation Accuracy	0.43
Validation Loss	2.31
Epoch	320
Total Training Time	2.66 H
Finish Date	17/12/2020

13. Resources.

Zhang, J. L., Liu, P., Zhang, F., & Song, Q. Q. (2018). CloudNet: Ground-based Cloud Classification with Deep Convolutional Neural Network. Geophysical Research Letters, 45. https://doi.org/10.1029/2018GL077787

Zhuang, J., Tang, T., Ding, Y., Tatikonda, S., Dvornek, N., Papademetris, X., & S. Duncan, J. (2020). AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. *Conference on Neural Information Processing Systems*. Retrieved from https://arxiv.org/abs/2010.07468

Pedgley, D. E. (2003). Luke Howard and his clouds. *Weather, 58*(2), 51-55. doi:10.1256/wea.157.02

Cloud Types and Precipitation. (n.d.). Retrieved November 21, 2020, from http://www.bom.gov.au/weather-services/about/cloud/cloud-types.shtml

Al Kamali. (2017, December 3). فـطنة و دهاء العرب في توقـع قدوم الـمـطر في ظل غياب التطور العلمي - الشيخ سعيد الكملي [Video]. YouTube. https://youtu.be/QcfY1lDMPKE