Weather prediction

Definition

domain background and project overview

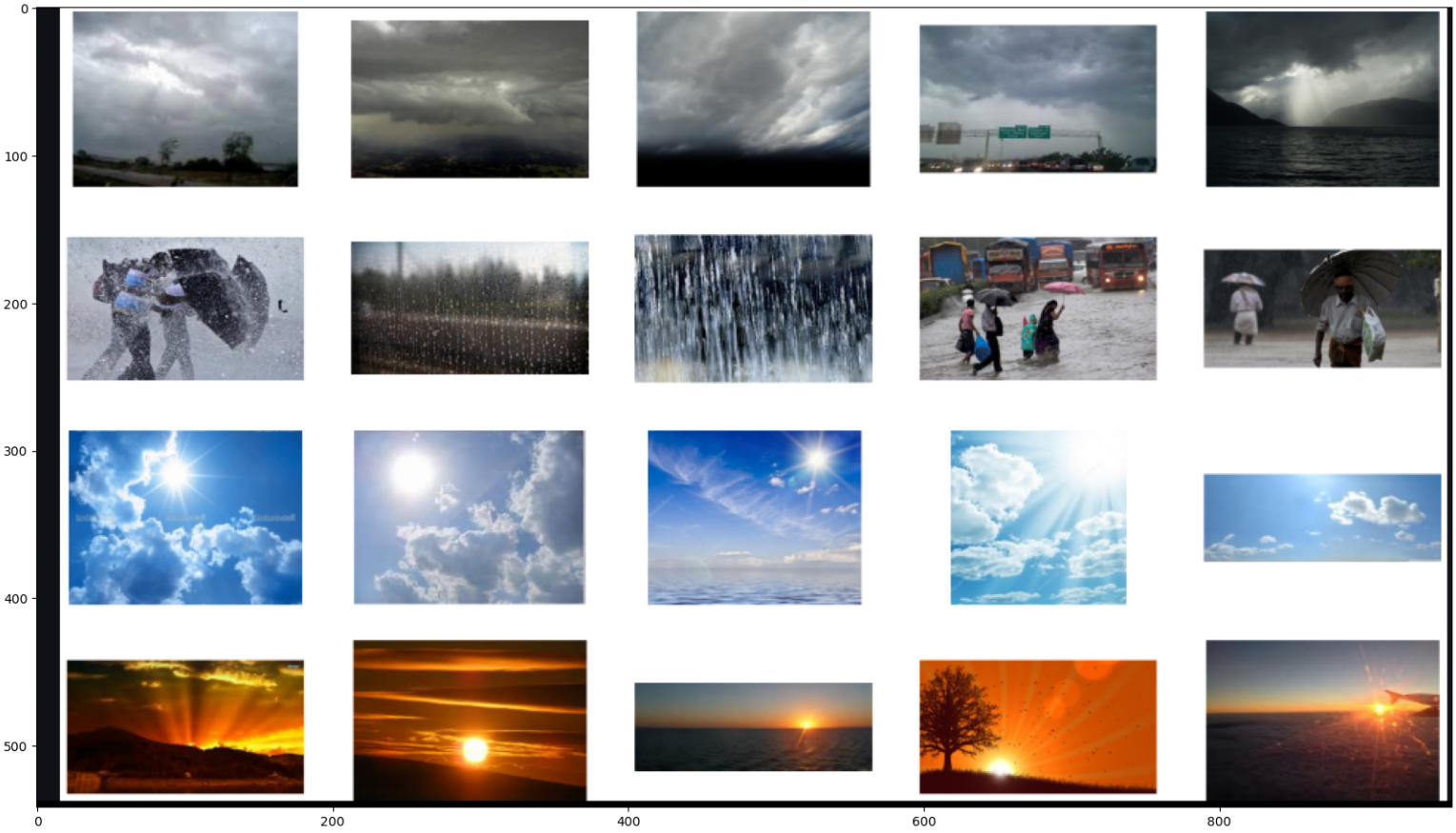
weather prediction is considered one of most importrant thing that we can face it daily and also most of our daily things than we can do will affect and also pepole will always know how weather is today to decide if they can go outside home or not . and also for drones that fly and can affect bedly for weather and planes and also in virtual reality if we want to put sunglass that be smart and we put this application inside it to get information about weather

in this project i make a deep learning model that can take a pohoto that taken by user and predict what kind of weather and also deployed it and put it in serveless servicess such lambda so it can taken and put in any application that will depends on this isuua and here the site that address this problem link here that contaions histroical data and information for every image from name we can know type for example siny12.png , cloudy11.png and so on

domin task

it is about multi-class weather recognition project, this dataset features 1125 images that are divided into four separate categories based on sunrise, cloudy, rainy, and sunshine.

this it considered as a problem from this image we can predict the type of weather and can cancel any thing that will affect badly on this issua .



as we can see threr are some of sample of dataset that used and here the site of this dataset

site of data

there are a four cast sunrise , cloudy, rain and shine and our task is to predict one of them according to image

problem statement

in this problem we can solve it by using deep learning models wich pretrained before this technique is called Transfer Learning so i use Resnet which is trained before on imagenet and make freeze for layer and add output layer that can fit data

the goal of network is to extract features and then use it and pass it through the fully connected layers which has an output layer that can predict result

we can put it in some of steps

1. collect data from site that i have mention
2. train a classifier that can know type of weather
3. deploy it and make an endpoint that can used for any application
4. put the endpoint in lambda so can handle traffic

solution statement

my solution is use transfer learning it will the best idea i choose resnet50 and train it on 90% of data and take about 10% to make validation

and also use some of data augumentation techniques to increase data and also make model train on some data that may be see them in the future

i make some of data processing such resize data and remove some of images not contains 3 channels so it considered as a not useful

describtion of solution to this problem

1. i use sagemaker aws services to train model
2. and then use sagemaker studio and specify instance type
3. first notebook is about data gathering , data cleaning and data visualization

Analysis

Data Explorations

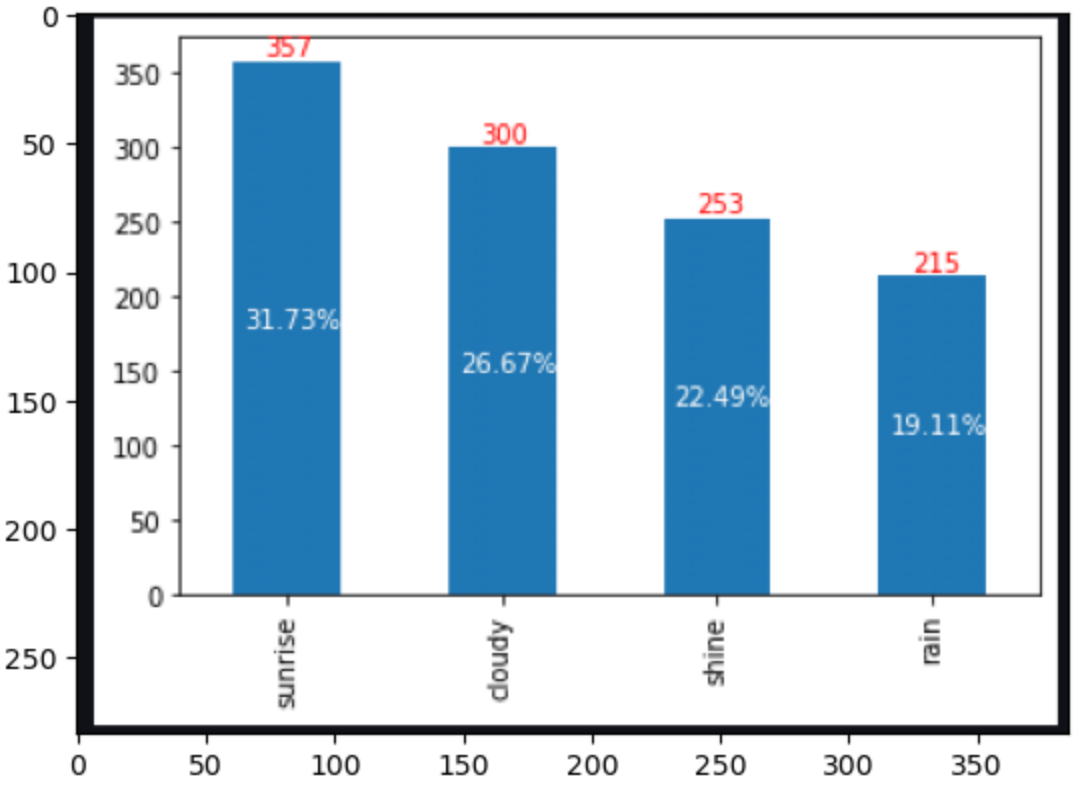
the weather Recognantion data consists of 1125 images that are divided into four separate categories based on sunrise, cloudy, rainy, and sunshine.

375 image for sunrise , 300 for cloudy , 253 for shine and 219 for rain it not enough more but transfer learning may help

data set came for different shape and alos there are a few image not compatable for others so i remove it

and make resize for all data to specifiic size is 300 for height and 500 for width and all have 3 channels

after that make some of data augumentation such fliping or crop



and this is a count and frequency for every class that used it consider is low so we need to collect more

Link of Dataset

Data preprocessing

data preprocessing done in collect and visulaize data notebook

- first collect data and extract files to make data suitable for reading and preprocessing
- ectract lable from name of image by usig reguler expression
- then collect data in dataframe
- randomized data on data frame
- make stratified split of data
- split data into 90% train and 10% validation
- save trian dataframe into train folder by move all images on image columns in dataframe to train folder
- move all images on image column on valiadation dataframe into valid folder
- apply mapping on label to convert it from string to integer maps = {'sunrise': 0, 'cloudy': 1, 'rain': 2, 'shine': 3}

more data informations

data used in imerit site

this dataset features 1125 images that are divided into four separate categories based on sunrise, cloudy, rainy, and sunshine.

we can download ot from this link

Multi-class weather dataset(MWD) for image classification is a valuable dataset used in the research paper entitled “Multi-class weather recognition from still image using heterogeneous ensemble method”. The dataset provides a platform for outdoor weather analysis by extracting various features for recognizing different weather conditions.

label for data it is name of image itsel such cloudy1.png , shine101.png and so on

the image is RGB and average size is 200 , 500 and have 3 channels

the data is collected from scraping some data on the interenet and we can make more scraping to get more data if we have ennough time or if model needed

the data will be normalized and rescaled to make pixels in range 0 , 1 insted of 0 , 255 and then normalize image by subtract every pixel in every channel from specific number normally close to 0.5 and also divide the result by std normally closer to 0.3 all of data is 3 channels except two image that not have 3 channels so i droped it

benchmark model

there are more models that can handle this well but in our case i haven't much data so we can collect data to increase performane but performance of data on this size it consider a good

most of data available or solution on the internet in kaggle for example it have data in shape of table that contain for example tempreture or some of measurement that not available to all person to measure this but in world of smart phones we can use camara insted of this measuremnets to know type of weather insted it .

so i build a machine learning model that can predict it well by only photo of sky

i try to to use transfer learning because data are a little and not used also in fune tuning it need data more than i have so i use resnet50 it suitable for this sutation and also trained on imagenet data that have the same features that my model will trained on

metrics that i have to perfoemance model

it take a few time to respone image and i deploy two endpoint one to take numpy arry and anthor to take PIL image to give varaiaty it we use one of them

time taken from model to response is a little it less than 200ms so it will achieve real-time response

i use this data to build most suitable solution in macine learning to use CNN it will be good in this sutation it can capture information and pass it to fully connected layers to make decision boundery and can make classifications

accuarcy will improve in validation a little so we need more data or chose model that more simpler than one is used

accuracy on train that incresed untill it reach 90% and more so we have overfiting but we can handle it by adding more data if we have more time and money and coputation power

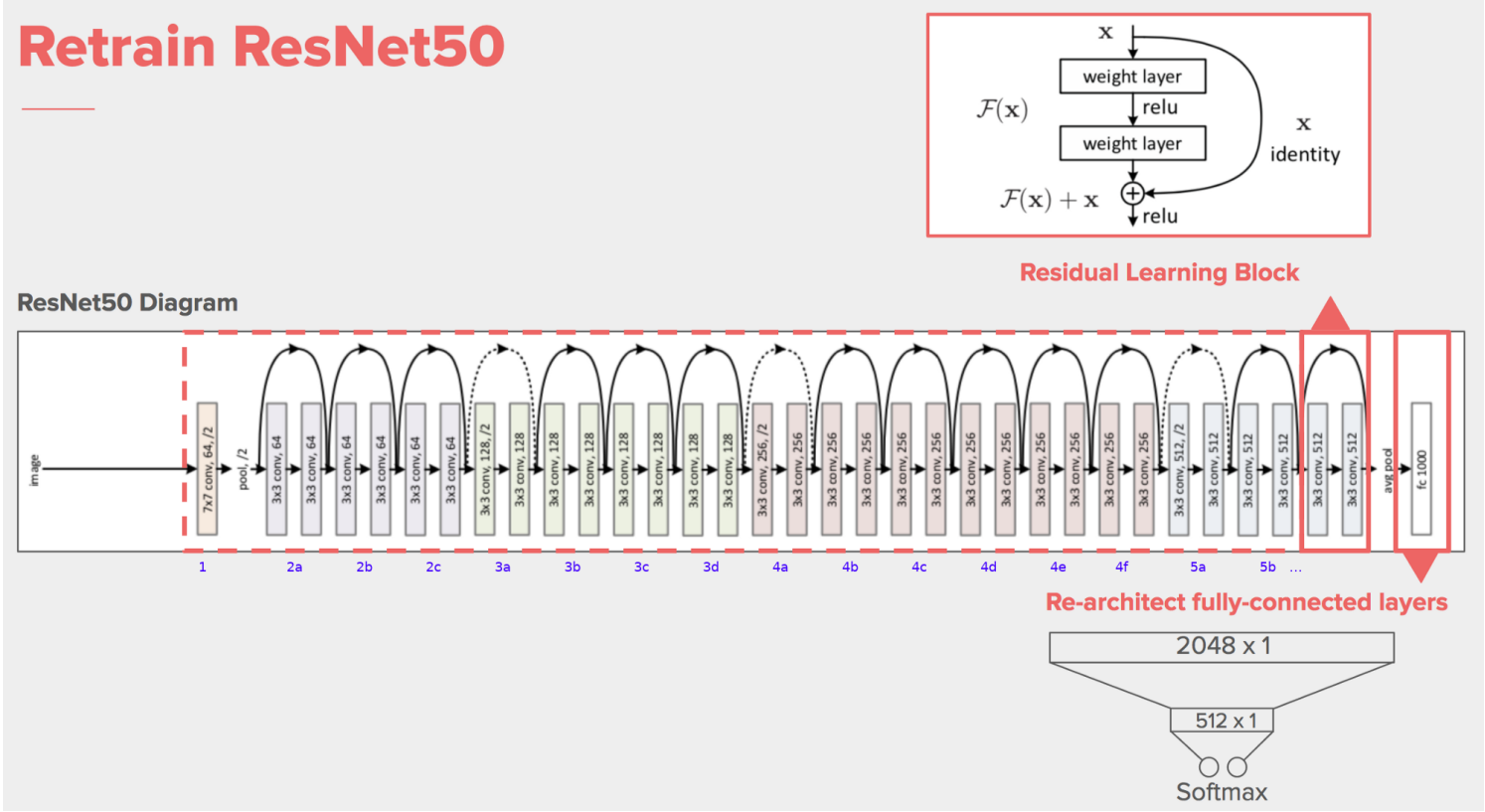
Evaulation Metrics

our problem is considered as a classification project so in this case i used a accuracy to see how many time that model can predict well among all dataset and also see loss function value our loss in this case is Categoricla Cross Entropy Loss it sufficient in multi class classification we can also use some metrics such precision or recall but our case i interset on all classes not speciified one so accuracy it useful

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Algorithmes

i used a pretrained model called resnet50



resnet is about deep learning model that used in computer vision

architecture of Resnet as follow

we can divide at as three parts

- part 1 is about blocks of CNN and Pooling layers and add batch-normalization layer to avoid vansishing and exploding readients
- and then we add the output for every block to input of it to avoid vanishing gradients
- part 2 is about adaptaive average pooling to reduce number of features that go in fully connected layers and also we make a classification mode so position of object don't take in calculation
- part3 is about FC layers that take input features and make a decision boundery to can classify object well

and finally i freeze all part1 and part2 layers that make feature extraction and trained only part3 (Fully connected layers to make predictions

Model implementation

- pass the batch of data in every time for very epoch data is batched to 64 images per batch and data first pass to transformes which make scaling and normalizing data
- and then pass go through the features layers to extract features and finally go to classification part to predict outcome
- compute loss function that take output of model which called model prediction and take also actual output of model
- after that we calc gradients vector to use it to make update for weights
- and then repeat this process in my case i reperet 60 times

Project refinement and improvement

i do some of issua in the first

i use vgg16 first and see ot give me poor performane so i decide to choose anthoe one more sufficient such resnet50

and then because of data fewer than expected i do some of data augumentations such Crop or fliping and so on

after that i make frezee for all feature extraction layer because data is small if it biger more we can make fune tuning

this technique help me to improve accuracy i tried more than pretrained model such vgg and squeznet and alos add more layers in clasifier part but all not imroved more such resnet50 and only add one layer in output

we can improve accuracy if we collect more data and add more layers with dropout or make regularization as i used in my network

Model evaluation and model performance

as we see from this result blow we can see that accuracy in training data is good and it enhance after epoch from training and if we see the accuracy of validation we see that the accuracy not improved well and there are a big gap in training accuracy and validation accuracy that indicets that there are a overfit on the model

one of solution we can do in this case and because of data is small than expected is to collect more data to solve this issua or try to choose anthore pretrained model may be increase accuracy of validation

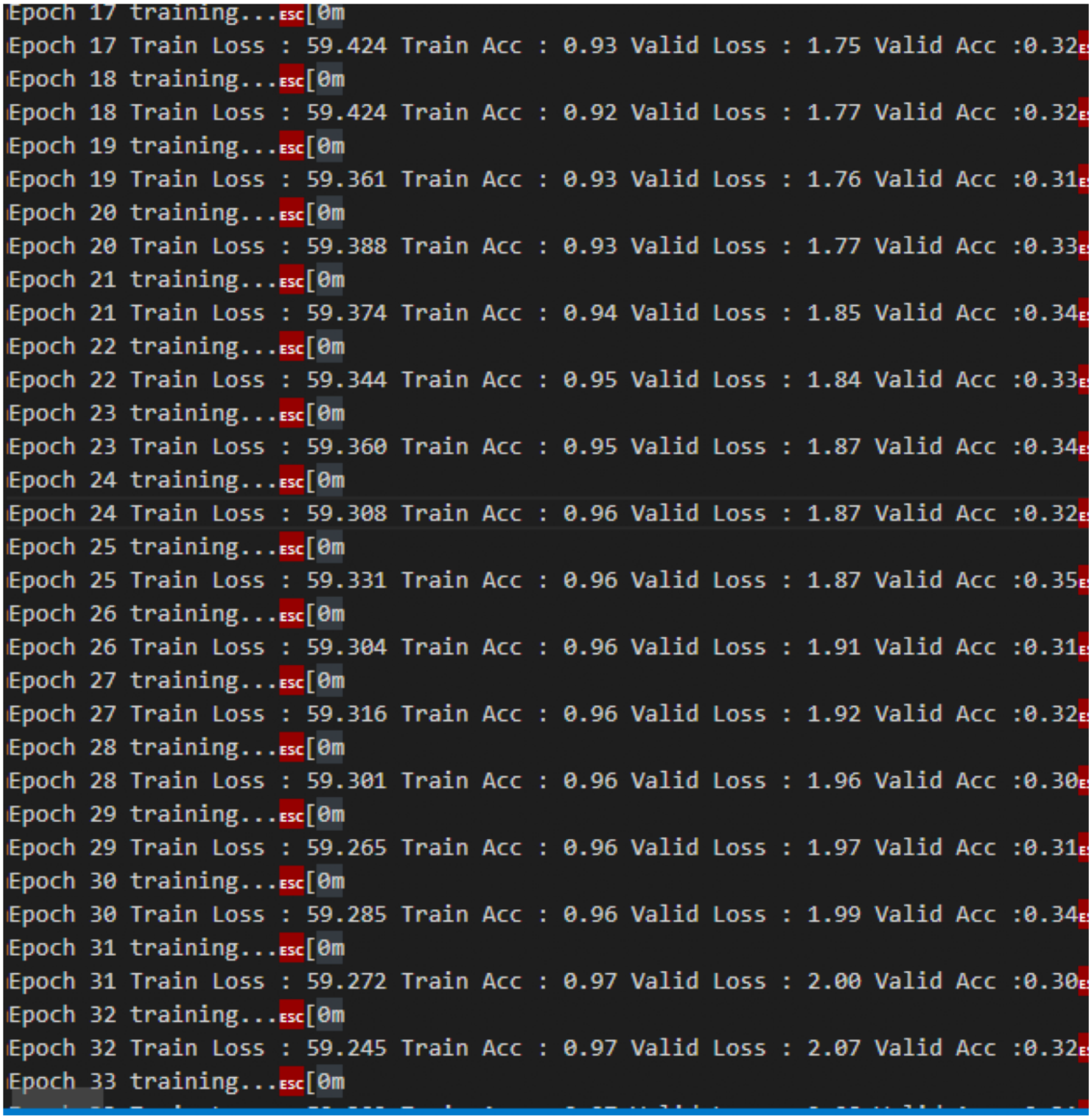
i use this data to build most suitable solution in macine learning to use CNN it will be good in this sutation it can capture information and pass it to fully connected layers to make decision boundery and can make classifications

accuarcy will improve in validation a little so we need more data or chose model that more simpler than one is used

accuracy on train that incresed untill it reach 90% and more so we have overfiting but we can handle it by adding more data if we have more time and money and coputation power

```
Epoch 1 Train Loss : 60.402 Train Acc : 0.25 Valid Loss : 1.37 Valid Acc :0.32
Epoch 2 training...[0m
Epoch 2 Train Loss : 60.143 Train Acc : 0.32 Valid Loss : 1.40 Valid Acc :0.32
Epoch 3 training...[0m
Epoch 3 Train Loss : 59.983 Train Acc : 0.30 Valid Loss : 1.55 Valid Acc :0.32
Epoch 4 training...[0m
Epoch 4 Train Loss : 59.899 Train Acc : 0.31 Valid Loss : 1.68 Valid Acc :0.35
Epoch 5 training...[0m
Epoch 5 Train Loss : 59.830 Train Acc : 0.31 Valid Loss : 1.59 Valid Acc :0.30
Epoch 6 training...[0m
Epoch 6 Train Loss : 59.763 Train Acc : 0.38 Valid Loss : 1.55 Valid Acc :0.32
Epoch 7 training...[0m
Epoch 7 Train Loss : 59.719 Train Acc : 0.56 Valid Loss : 1.53 Valid Acc :0.31
Epoch 8 training...[0m
Epoch 8 Train Loss : 59.677 Train Acc : 0.76 Valid Loss : 1.55 Valid Acc :0.30
Epoch 9 training...[0m
Epoch 9 Train Loss : 59.594 Train Acc : 0.84 Valid Loss : 1.60 Valid Acc :0.28
Epoch 10 training...[0m
Epoch 10 Train Loss : 59.578 Train Acc : 0.86 Valid Loss : 1.61 Valid Acc :0.33
Epoch 11 training...[0m
Epoch 11 Train Loss : 59.573 Train Acc : 0.87 Valid Loss : 1.64 Valid Acc :0.31
Epoch 12 training...[0m
Epoch 12 Train Loss : 59.555 Train Acc : 0.88 Valid Loss : 1.66 Valid Acc :0.34
Epoch 13 training...[0m
Epoch 13 Train Loss : 59.521 Train Acc : 0.90 Valid Loss : 1.69 Valid Acc :0.32
Epoch 14 training...[0m
Epoch 14 Train Loss : 59.465 Train Acc : 0.90 Valid Loss : 1.73 Valid Acc :0.34
Epoch 15 training...[0m
Epoch 15 Train Loss : 59.448 Train Acc : 0.91 Valid Loss : 1.71 Valid Acc :0.31
Epoch 16 training...[0m
Epoch 16 Train Loss : 59.418 Train Acc : 0.92 Valid Loss : 1.72 Valid Acc :0.31
Epoch 17 training...[0m
Epoch 17 Train Loss : 59.424 Train Acc : 0.93 Valid Loss : 1.75 Valid Acc :0.32
```

```
Epoch 17 training...[0m
Epoch 17 Train Loss : 59.424 Train Acc : 0.93 Valid Loss : 1.75 Valid Acc :0.32
Epoch 18 training...[0m
Epoch 18 Train Loss : 59.424 Train Acc : 0.92 Valid Loss : 1.77 Valid Acc :0.32
Epoch 19 training...[0m
Epoch 19 Train Loss : 59.361 Train Acc : 0.93 Valid Loss : 1.76 Valid Acc :0.31
Epoch 20 training...[0m
Epoch 20 Train Loss : 59.388 Train Acc : 0.93 Valid Loss : 1.77 Valid Acc :0.33
Epoch 21 training...[0m
Epoch 21 Train Loss : 59.374 Train Acc : 0.94 Valid Loss : 1.85 Valid Acc :0.34
Epoch 22 training...[0m
Epoch 22 Train Loss : 59.344 Train Acc : 0.95 Valid Loss : 1.84 Valid Acc :0.33
Epoch 23 training...[0m
Epoch 23 Train Loss : 59.360 Train Acc : 0.95 Valid Loss : 1.87 Valid Acc :0.34
Epoch 24 training...[0m
Epoch 24 Train Loss : 59.308 Train Acc : 0.96 Valid Loss : 1.87 Valid Acc :0.32
Epoch 25 training...[0m
Epoch 25 Train Loss : 59.331 Train Acc : 0.96 Valid Loss : 1.87 Valid Acc :0.35
Epoch 26 training...[0m
Epoch 26 Train Loss : 59.304 Train Acc : 0.96 Valid Loss : 1.91 Valid Acc :0.31
Epoch 27 training...[0m
Epoch 27 Train Loss : 59.316 Train Acc : 0.96 Valid Loss : 1.92 Valid Acc :0.32
Epoch 28 training...[0m
Epoch 28 Train Loss : 59.301 Train Acc : 0.96 Valid Loss : 1.96 Valid Acc :0.30
Epoch 29 training...[0m
Epoch 29 Train Loss : 59.265 Train Acc : 0.96 Valid Loss : 1.97 Valid Acc :0.31
Epoch 30 training...[0m
Epoch 30 Train Loss : 59.285 Train Acc : 0.96 Valid Loss : 1.99 Valid Acc :0.34
Epoch 31 training...[0m
Epoch 31 Train Loss : 59.272 Train Acc : 0.97 Valid Loss : 2.00 Valid Acc :0.30
Epoch 32 training...[0m
Epoch 32 Train Loss : 59.245 Train Acc : 0.97 Valid Loss : 2.07 Valid Acc :0.32
Epoch 33 training...[0m
```



Poject Design

1. upload data from interenet (the link that i have mention)
2. extract zip files data in local dir in sagemaker
3. make data preprocessing and visulaization
4. remove some files that have isuua and not releated to the majority of data
5. assert that data is more prepred to get in modeling stage
6. upload data from sagemaker to s3
7. make test on data in sagemaker studio before make a training script
8. write a script that will train network

9. upload a pretrained model and freeze features extraction layer
10. customize model on our problem and dataset
11. build a transformers for trainining and validation data
12. make monitor on data to see cpubootelenck and vansishing gradients , overfitting
13. Estaplish a Scrpit mode and pass for it out script
14. make debugger and Profiler to monior model during training
15. train model for 50 epochs
16. write a script for interface
17. deploy model one for accept PIL image
18. make endpoint that acceepts PIL image
19.
20. make anthor endpoint to accept numpy image
21.
22. create a lambda function for our endpoint
23.
24. confiure lambda to can handle and access sagemaker and endpoint

```
[NbConvertApp] Converting notebook test.ipynb to webpdf
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 3139544 bytes to test.pdf
```