



# Project 15

## Group 1



## Objective - Goal

- The goal of our project is to use cloud based Azure IoT Edge and ML/AI Cloud based technology to train, test and deploy AI models to an IoT Edge device
- The device is capable of detecting both elephant and gunshot sounds at a defined locations with the specific purpose of catching individuals who are killing elephants for profit

# Background - Project Preamble - Group

Geographic Extent: latitude 1.89020 to 2.32474, longitude 16.44315 to 16.74016 (WGS84)

Site	Habitat (at recorder site)	Protection
nn01a	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn01c	mixed forest-closed	national park
nn01d	mixed forest-closed	national park
nn01e	mixed forest-closed	national park
nn01f	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn01g	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn02a	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn02b	mixed forest-closed	national park
nn02c	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn02d	mixed forest-closed	national park
nn02e	mixed forest-closed	national park
nn02f	mixed forest-closed	national park
nn02g	mixed forest-open	national park
nn03a	mixed forest-closed	national park
nn03b	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn03c	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn03d	mixed forest-closed	national park
nn03e	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	national park
nn03f	swamp forest	national park
nn03g	mixed forest-closed	national park
nn04a	foret mixte	national park
nn04b	mixed forest-closed	national park
nn04c	mixed forest-closed	national park
nn04d	Gilbertiodendron (mono-dominant); very tall canopy, sparse understory	logging – active 2017-2018
nn04e	mixed forest-open	logging – active 2017-2018
nn04f	mixed forest-closed	logging – active 2017-2018
nn05a	mixed forest-closed	national park
nn05b	mixed forest-closed	national park
nn05c	mixed forest-closed	logging – active 2017-2018
nn05d	mixed forest-closed	logging – active 2017-2018
nn05e	swamp forest	logging – active 2017-2018
nn05f	mixed forest-closed	logging – active 2017-2018

Challenges to forensic analysis of gunshots ... due to broadcast and noisy nature of acoustic channel.

Gunshots are a result of multiple acoustic events, namely:

- Muzzle blast (explosion inside barrel)
- Ballistic shockwave (supersonic projectile)

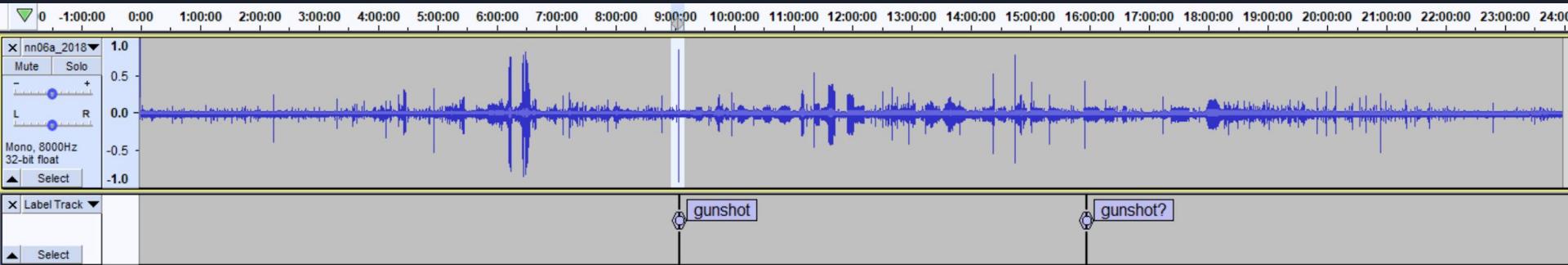
Most ad-hoc models do not address ‘Room Reverb’, ‘Concert Reverb’, and doppler effect.

Raponi et al. Sounds of Guns:

# Analysis of problem

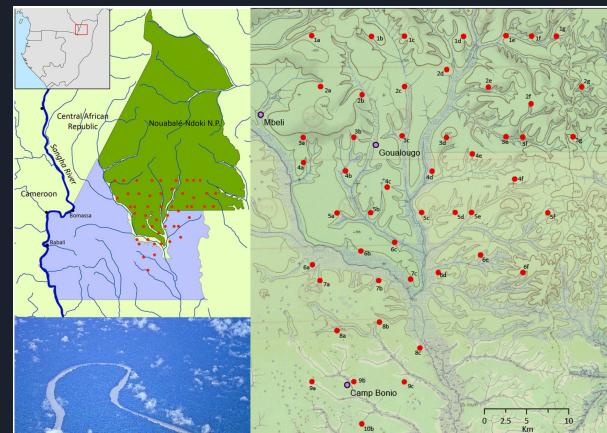
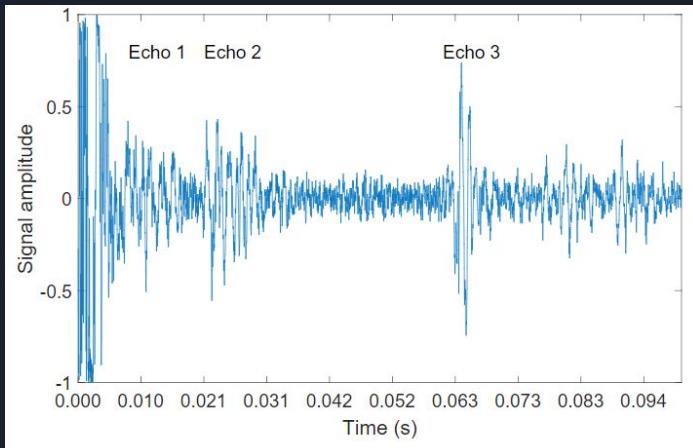
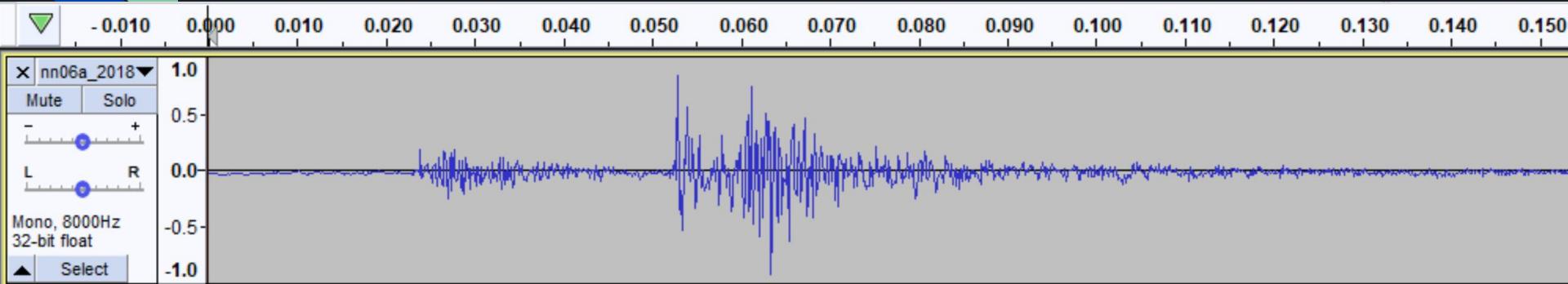
First task to analyse data (ELP audios). Some questions to ask:

- What does a gunshot sound 'look' like in the 24 hr audio data gathered.
- Does it look similar to literature search of 'domesticated' gunsounds.



# Data preview

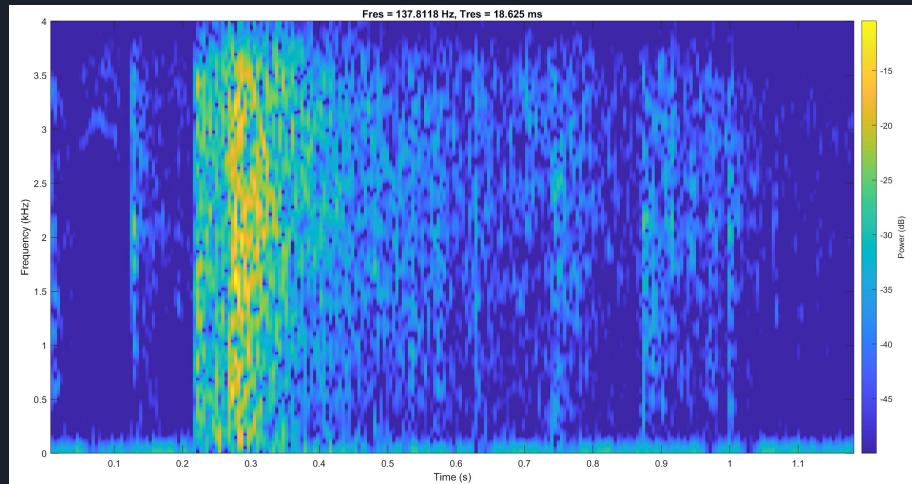
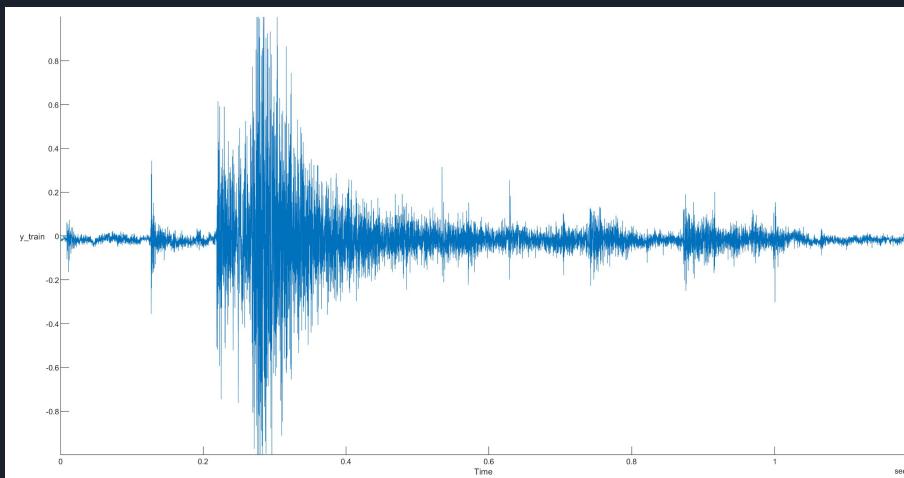
Does it look similar to literature search of 'domesticated' gunsounds.



# Data preview

From power spectrogram we can see that a muzzle blast has a large frequency range and differs between gun types. This may be a challenge to filter data.

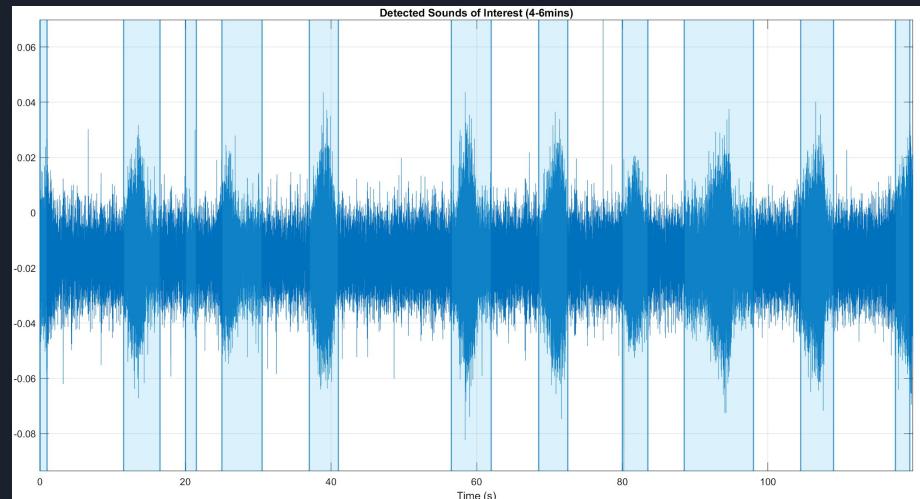
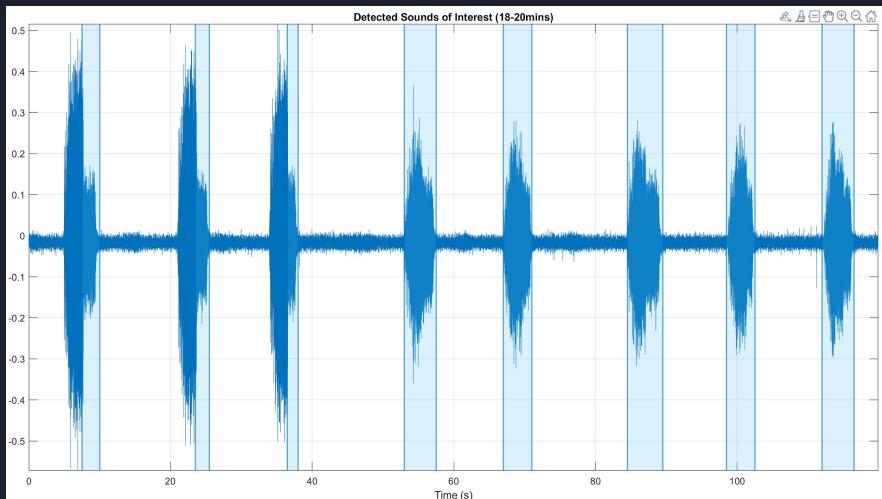
The question now is how best to 'harvest' this data in the large ELP audios.



# Data preview leading to a MATLAB data processing approach

MATLAB function to the 'detectSpeech' to find audio snippets of interest. This will be stored as a datastore.

This method is to only analyse / compare the audios against a trained audio of a gunshot. This is lighten the load as the entire audio ELP data will not have to be analysed but only snippets of interest.





# Observation

Observation 1: Acoustic signature of guns are distinctive based on literature, unique to even gun type. This can be used to help label audio snippets for gun sounds.

Observation 2: However echo characteristics differ than those in the wild as gun shots are not cultivated in audio-rooms or urban environments (3ms vs 60ms).

Observation 3: Frequency range of gunshots are larger (100-4000Hz), bandwidth filters should be explored. (0-24KHz).

Observation 4: From literature search, model may require human-assisted supervised learning from ELP.

Observation 5: Validity of using purely cultivated gunshot sounds to detect gunshot sounds in wild (ELP data) is questioned?

# Background - Project Preamble - Group

The task is to create and implement that allows detecting gunshots. The problem is that there are few data points and also these are very rare sounds. We were able to identify two alternatives to solve this problem.

Create a shot classification model using:

Use the actual recordings and then identify gunshots.

- Computationally expensive when processing and analyzing all audios files (24 hours of audio in each recording)

Use an external database with gunshot sounds

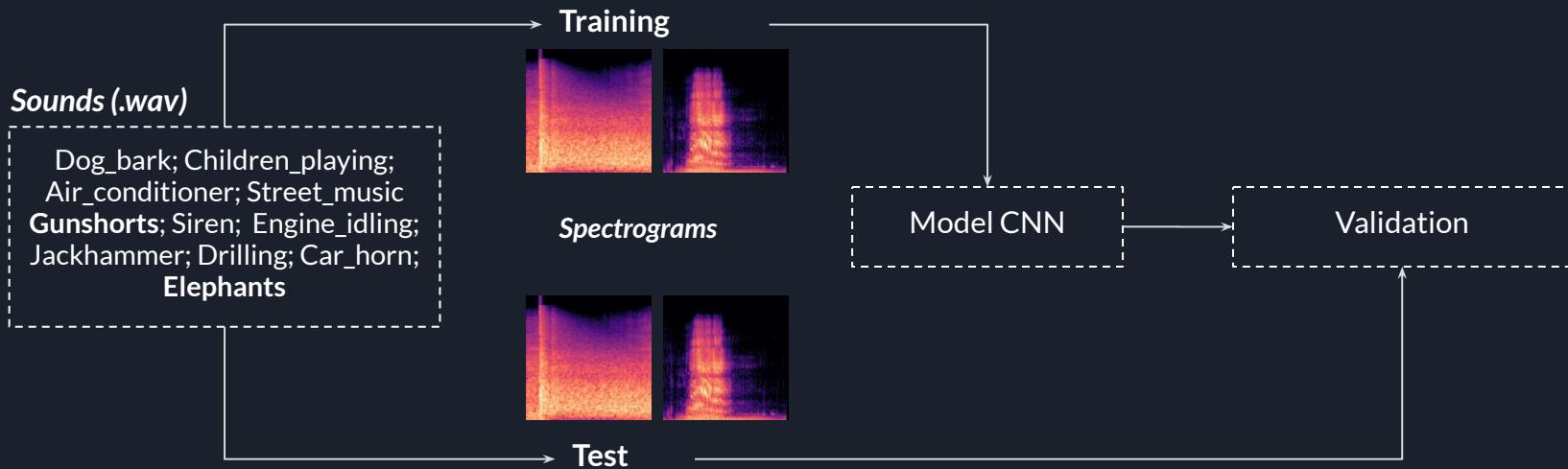
- Shots from external databases do not have ambient sound. There is a risk that the model cannot identify actual gunshots.

Deploy the model in a cloud environment

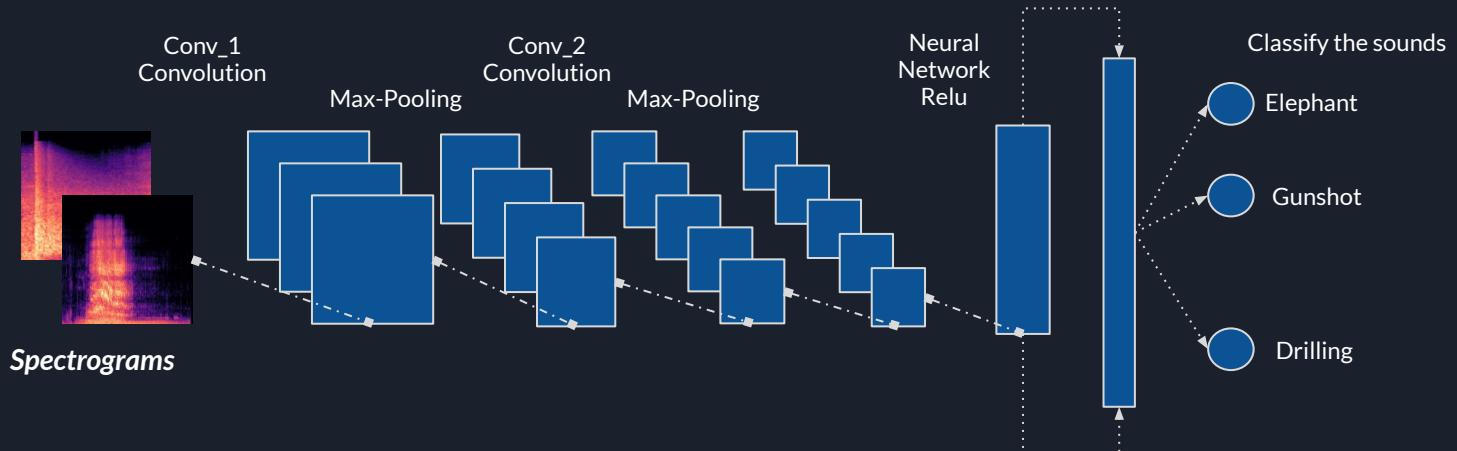
Edge device

# Data structure and model flow

We decided to use an external database that has different types of sounds (10 classes) including gunshots. Additionally, we also add elephant sounds.



# Model structure

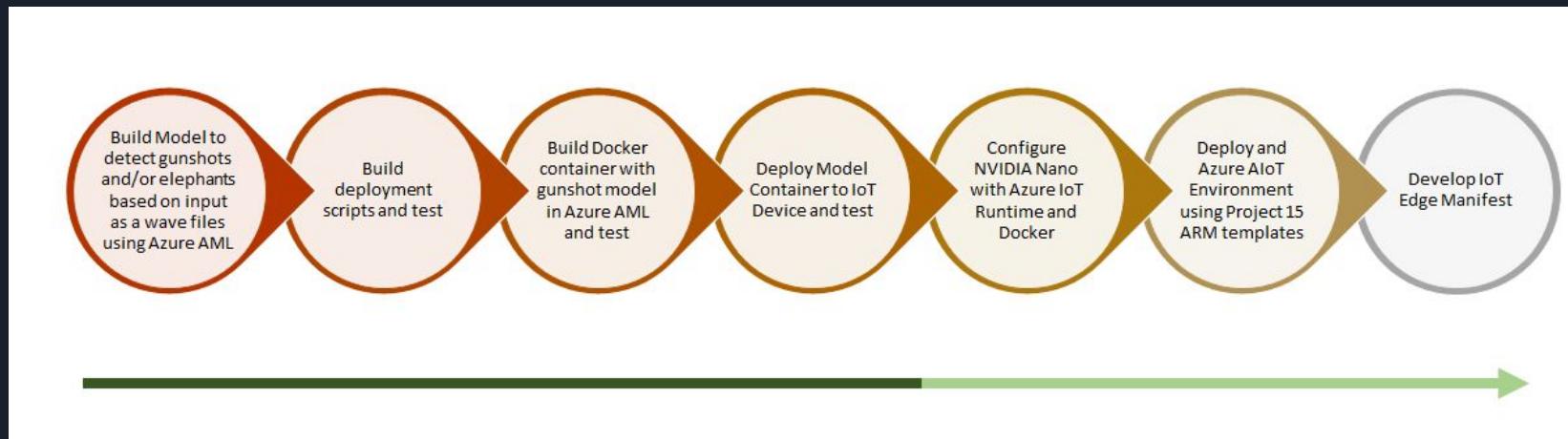




# Assumptions - Group

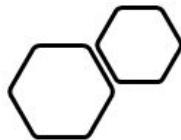
- During the model training we are making the assumption that we can use a large corpus of urban gunshot wav files to train a model to detect gunshot sounds that would occur in areas that are very different than those in urban areas
- We are assuming that the IoT edge device will have connectivity either with a cellular connection or by a satellite connect neither of which would be continuous i.e. the edge device could experience periods without connectivity.
- The edge device will have access either via the cell or satellite connection to lat/long coordinates or would have them as a device twin property updated from the hub once the device was in position (the latter being the least complicated configuration)

# Project 15 | End to end workflow





Build Models to detect gunshots or elephant sounds using wav files



# Methodology

## Model Step by Step Process:

1. Convert the train/test audio files into train/test spectrogram image files
2. Convert image pixels to NumPy arrays and then normalize the arrays
3. Create the Convolutional Base
4. Compile and Train the model
5. Evaluate the model

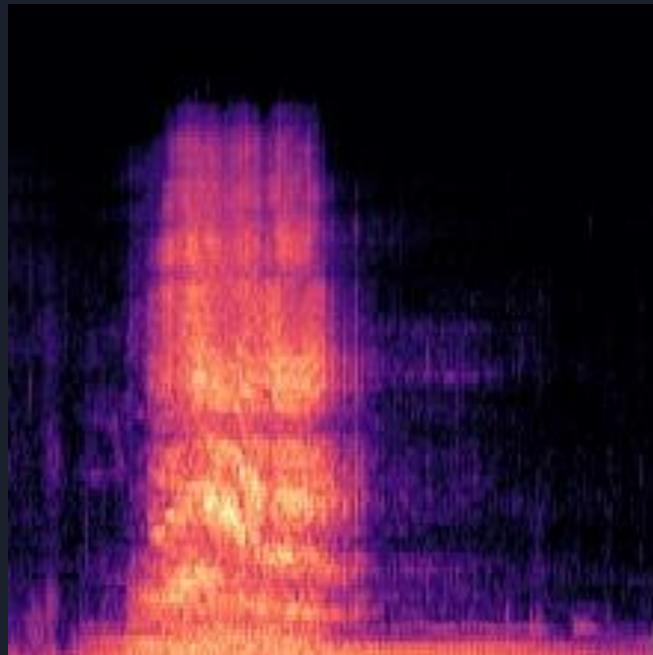


# Discussion

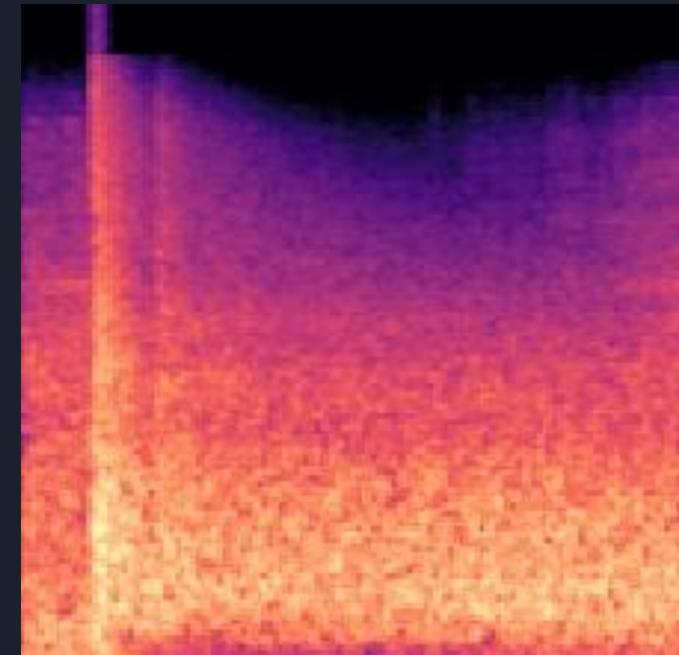
## Approach:

- Audio files ----> Spectrogram images ----> pixel arrays ---->  
Normalization ----> Train/Evaluate model
- Spectrograms:
  - Visualize frequencies present in audio recordings
  - Frequency patterns vary for each class of sound
  - Classify images based on the frequencies

Elephant sound

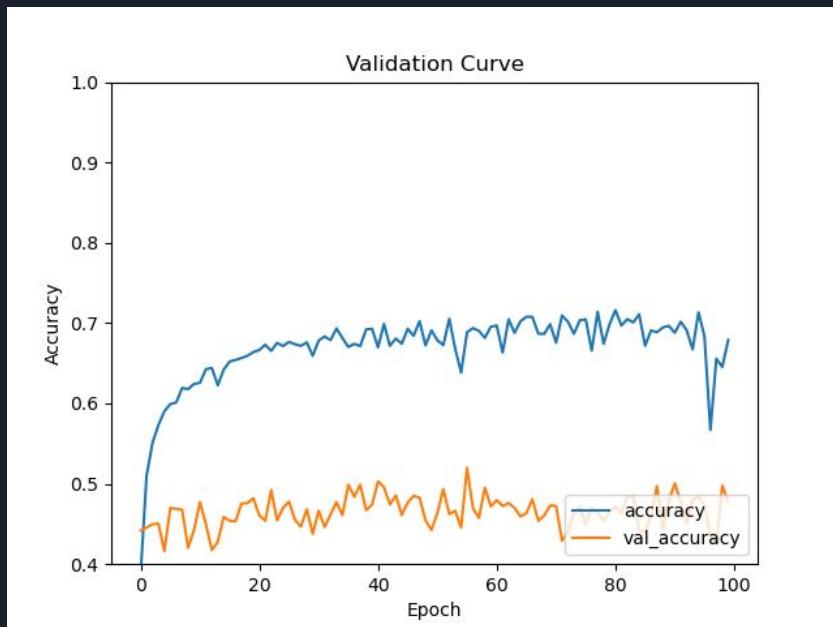


Gunshot sound

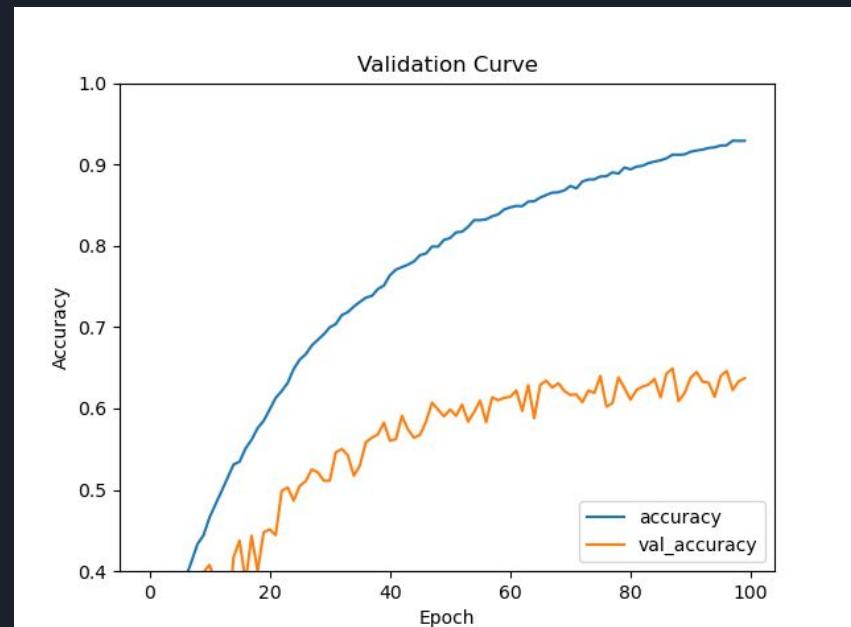


# Discussion - Hyperparameter tuning

Adam Optimizer



SGD Optimizer



# Adam

=====			
-----MODEL PREDICTIONS-----			
=====			
0	135526-6-0-0.wav	gun_shot	gun_shot
1	135526-6-1-0.wav	gun_shot	street_music
2	135526-6-2-0.wav	gun_shot	gun_shot
3	135526-6-3-0.wav	gun_shot	gun_shot
4	135526-6-4-0.wav	gun_shot	gun_shot
5	135526-6-5-0.wav	gun_shot	gun_shot
6	135526-6-6-0.wav	gun_shot	gun_shot
7	135526-6-7-0.wav	gun_shot	gun_shot
8	135526-6-8-0.wav	gun_shot	gun_shot
9	135526-6-9-0.wav	gun_shot	street_music
10	<u>203929-7-3-10.wav</u>	jackhammer	jackhammer
11	203929-7-3-2.wav	jackhammer	jackhammer
12	203929-7-3-3.wav	jackhammer	jackhammer
13	203929-7-3-5.wav	jackhammer	jackhammer
14	203929-7-3-9.wav	jackhammer	jackhammer
15	344-3-0-0.wav	dog_bark	dog bark
16	344-3-1-0.wav	dog_bark	dog bark
17	344-3-4-0.wav	dog_bark	dog bark
18	344-3-5-0.wav	dog_bark	dog bark
19	518-4-0-0.wav	drilling	drilling
20	518-4-0-1.wav	drilling	drilling
21	518-4-0-2.wav	drilling	jackhammer
22	518-4-0-3.wav	drilling	jackhammer
23	Elephant.1.wav	Elephant	street music
24	Elephant.10.wav	Elephant	air_conditioner
25	Elephant.11.wav	Elephant	Elephant
26	Elephant.12.wav	Elephant	street music
27	Elephant.13.wav	Elephant	children_playing
28	Elephant.14.wav	Elephant	dog bark
29	Elephant.15.wav	Elephant	air conditioned
30	Elephant.16.wav	Elephant	air conditioner
31	Elephant.17.wav	Elephant	air conditioner
32	Elephant.18.wav	Elephant	air conditioner
33	Elephant.19.wav	Elephant	Elephant
34	Elephant.2.wav	Elephant	street music
35	Elephant.20.wav	Elephant	engine_idling
36	Elephant.27.wav	Elephant	street music
37	Elephant.3.wav	Elephant	street music

This model has predicted 21/38 correctly, which is 55.26%  
Time of execution: 527.9309256076813

# SGD

=====			
-----MODEL PREDICTIONS-----			
=====			
0	135526-6-0-0.wav	gun_shot	gun_shot
1	135526-6-1-0.wav	gun_shot	gun_shot
2	135526-6-2-0.wav	gun_shot	gun_shot
3	135526-6-3-0.wav	gun_shot	gun_shot
4	135526-6-4-0.wav	gun_shot	gun_shot
5	135526-6-5-0.wav	gun_shot	gun_shot
6	135526-6-6-0.wav	gun_shot	gun_shot
7	135526-6-7-0.wav	gunshot	gunshot
8	135526-6-8-0.wav	gun_shot	gun_shot
9	135526-6-9-0.wav	gunshot	gunshot
10	<u>203929-7-3-10.wav</u>	jackhammer	jackhammer
11	203929-7-3-2.wav	jackhammer	jackhammer
12	203929-7-3-3.wav	jackhammer	jackhammer
13	203929-7-3-5.wav	jackhammer	jackhammer
14	203929-7-3-9.wav	jackhammer	jackhammer
15	344-3-0-0.wav	dog bark	dog bark
16	344-3-1-0.wav	dog bark	dog bark
17	344-3-4-0.wav	dog bark	dog bark
18	344-3-5-0.wav	dog bark	dog bark
19	518-4-0-0.wav	drilling	drilling
20	518-4-0-1.wav	drilling	drilling
21	518-4-0-2.wav	drilling	drilling
22	518-4-0-3.wav	drilling	drilling
23	Elephant.1.wav	Elephant	Elephant
24	Elephant.10.wav	Elephant	drilling
25	Elephant.11.wav	Elephant	Elephant
26	Elephant.12.wav	Elephant	Elephant
27	Elephant.13.wav	Elephant	Elephant
28	Elephant.14.wav	Elephant	Elephant
29	Elephant.15.wav	Elephant	Elephant
30	Elephant.16.wav	Elephant	Elephant
31	Elephant.17.wav	Elephant	Elephant
32	Elephant.18.wav	Elephant	Elephant
33	Elephant.19.wav	Elephant	Elephant
34	Elephant.2.wav	Elephant	Elephant
35	Elephant.20.wav	Elephant	Elephant
36	Elephant.27.wav	Elephant	Elephant
37	Elephant.3.wav	Elephant	Elephant

This model has predicted 37/38 correctly, which is 97.37%  
Time of execution: 550.3758449554443

@ 100  
Epochs

# Epoch size = 100

-----MODEL PREDICTIONS-----				
Filename	Actual_Result	Model_Predicted_Result		
0 135526-6-0-0.wav	gun_shot	gun_shot		
1 135526-6-1-0.wav	gun_shot	gun_shot		
2 135526-6-2-0.wav	gun_shot	gun_shot		
3 135526-6-3-0.wav	gun_shot	gun_shot		
4 135526-6-4-0.wav	gun_shot	gun_shot		
5 135526-6-5-0.wav	gun_shot	gun_shot		
6 135526-6-6-0.wav	gun_shot	gun_shot		
7 135526-6-7-0.wav	gun_shot	gun_shot		
8 135526-6-8-0.wav	gun_shot	gun_shot		
9 135526-6-9-0.wav	gun_shot	gun_shot		
10 203929-7-3-10.wav	jackhammer	jackhammer		
11 203929-7-3-2.wav	jackhammer	jackhammer		
12 203929-7-3-3.wav	jackhammer	jackhammer		
13 203929-7-3-5.wav	jackhammer	jackhammer		
14 203929-7-3-9.wav	jackhammer	jackhammer		
15 344-3-0-0.wav	dog_bark	dog_bark		
16 344-3-1-0.wav	dog_bark	dog_bark		
17 344-3-4-0.wav	dog_bark	dog_bark		
18 344-3-5-0.wav	dog_bark	dog_bark		
19 518-4-0-0.wav	drilling	drilling		
20 518-4-0-1.wav	drilling	drilling		
21 518-4-0-2.wav	drilling	drilling		
22 518-4-0-3.wav	drilling	drilling		
23 Elephant.1.wav	Elephant	Elephant		
24 Elephant.10.wav	Elephant	drilling		
25 Elephant.11.wav	Elephant	Elephant		
26 Elephant.12.wav	Elephant	Elephant		
27 Elephant.13.wav	Elephant	Elephant		
28 Elephant.14.wav	Elephant	Elephant		
29 Elephant.15.wav	Elephant	Elephant		
30 Elephant.16.wav	Elephant	Elephant		
31 Elephant.17.wav	Elephant	Elephant		
32 Elephant.18.wav	Elephant	Elephant		
33 Elephant.19.wav	Elephant	Elephant		
34 Elephant.2.wav	Elephant	Elephant		
35 Elephant.20.wav	Elephant	Elephant		
36 Elephant.27.wav	Elephant	Elephant		
37 Elephant.3.wav	Elephant	Elephant		

This model has predicted 37/38 correctly, which is 97.37%  
Time of execution: 550.3758449554443

# Epoch size = 50

-----MODEL PREDICTIONS-----				
Filename	Actual_Result	Model_Predicted_Result		
0 135526-6-0-0.wav	gun_shot	gun_shot		
1 135526-6-1-0.wav	gun_shot	gun_shot		
2 135526-6-2-0.wav	gun_shot	gun_shot		
3 135526-6-3-0.wav	gun_shot	gun_shot		
4 135526-6-4-0.wav	gun_shot	gun_shot		
5 135526-6-5-0.wav	gun_shot	gun_shot		
6 135526-6-6-0.wav	gun_shot	gun_shot		
7 135526-6-7-0.wav	gun_shot	gun_shot		
8 135526-6-8-0.wav	gun_shot	gun_shot		
9 135526-6-9-0.wav	gun_shot	gun_shot		
10 203929-7-3-10.wav	jackhammer	jackhammer		
11 203929-7-3-2.wav	jackhammer	jackhammer		
12 203929-7-3-3.wav	jackhammer	jackhammer		
13 203929-7-3-5.wav	jackhammer	jackhammer		
14 203929-7-3-9.wav	jackhammer	jackhammer		
15 344-3-0-0.wav	dog_bark	dog_bark		
16 344-3-1-0.wav	dog_bark	dog_bark		
17 344-3-4-0.wav	dog_bark	dog_bark		
18 344-3-5-0.wav	dog_bark	dog_bark		
19 518-4-0-0.wav	drilling	drilling		
20 518-4-0-1.wav	drilling	engine_idling		
21 518-4-0-2.wav	drilling	engine_idling		
22 518-4-0-3.wav	drilling	jackhammer		
23 Elephant.1.wav	Elephant	Elephant		
24 Elephant.10.wav	Elephant	air_conditioned		
25 Elephant.11.wav	Elephant	Elephant		
26 Elephant.12.wav	Elephant	Elephant		
27 Elephant.13.wav	Elephant	dog_bark		
28 Elephant.14.wav	Elephant	Elephant		
29 Elephant.15.wav	Elephant	Elephant		
30 Elephant.16.wav	Elephant	Elephant		
31 Elephant.17.wav	Elephant	Elephant		
32 Elephant.18.wav	Elephant	Elephant		
33 Elephant.19.wav	Elephant	Elephant		
34 Elephant.2.wav	Elephant	Elephant		
35 Elephant.20.wav	Elephant	dog_bark		
36 Elephant.27.wav	Elephant	Elephant		
37 Elephant.3.wav	Elephant	Elephant		

This model has predicted 31/38 correctly, which is 81.58%  
Time of execution: 252.86861968040466

# Epoch size = 10

-----MODEL PREDICTIONS-----				
Filename	Actual_Result	Model_Predicted_Result		
0 135526-6-0-0.wav	gun_shot	gun_shot		
1 135526-6-1-0.wav	gun_shot	street_music		
2 135526-6-2-0.wav	gun_shot	jackhammer		
3 135526-6-3-0.wav	gun_shot	gun_shot		
4 135526-6-4-0.wav	gun_shot	gun_shot		
5 135526-6-5-0.wav	gun_shot	jackhammer		
6 135526-6-6-0.wav	gun_shot	gun_shot		
7 135526-6-7-0.wav	gun_shot	street_music		
8 135526-6-8-0.wav	gun_shot	street_music		
9 135526-6-9-0.wav	gun_shot	street_music		
10 203929-7-3-10.wav	jackhammer	jackhammer		
11 203929-7-3-2.wav	jackhammer	jackhammer		
12 203929-7-3-3.wav	jackhammer	jackhammer		
13 203929-7-3-5.wav	jackhammer	jackhammer		
14 203929-7-3-9.wav	jackhammer	jackhammer		
15 344-3-0-0.wav	dog_bark	dog_bark		
16 344-3-1-0.wav	dog_bark	dog_bark		
17 344-3-4-0.wav	dog_bark	dog_bark		
18 344-3-5-0.wav	dog_bark	dog_bark		
19 518-4-0-0.wav	drilling	jackhammer		
20 518-4-0-1.wav	drilling	engine_idling		
21 518-4-0-2.wav	drilling	drilling		
22 518-4-0-3.wav	drilling	jackhammer		
23 Elephant.1.wav	Elephant	Elephant		
24 Elephant.10.wav	Elephant	air_conditioner		
25 Elephant.11.wav	Elephant	Elephant		
26 Elephant.12.wav	Elephant	Elephant		
27 Elephant.13.wav	Elephant	dog_bark		
28 Elephant.14.wav	Elephant	Elephant		
29 Elephant.15.wav	Elephant	Elephant		
30 Elephant.16.wav	Elephant	Elephant		
31 Elephant.17.wav	Elephant	Elephant		
32 Elephant.18.wav	Elephant	Elephant		
33 Elephant.19.wav	Elephant	Elephant		
34 Elephant.2.wav	Elephant	Elephant		
35 Elephant.20.wav	Elephant	dog_bark		
36 Elephant.27.wav	Elephant	Elephant		
37 Elephant.3.wav	Elephant	Elephant		

This model has predicted 13/38 correctly, which is 34.21%  
Time of execution: 73.53798031806946

# Model optimization

- Dropout is not necessary.
- SGD has better generalization performance than ADAM.  
*SGD usually improves model performance slowly but could achieve higher test performance."*
- The sounds are quite different so the model is quite general. If the sounds were more identical to each other, ADAM would be the best option. Since it could quickly get out of the local minimums.
- The model learns fast. It is not necessary to use a small learning rate.

Optimizer	Epochs	Dropout	Accuracy
SGD	100	No	97%
SGD	200	No	100%
SGD	50	No	82%
SGD	50	Yes	74%
SGD	100	Yes	76%

Source: <https://proceedings.neurips.cc/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf>



## Discussion

### Challenges:

- Imbalance in dataset. Owing to smaller sample size of elephant audio files
- File size of dataset too large



Build deployment environment, scripts and edge devices

# Creation of the Azure ML/AI deployment environment to support Project 15

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes links for User, AI, Troubleshoot, Projects, Com, and Home, along with a search bar and a 'Not syncing' status message. The main title is 'Microsoft Azure' with a search bar below it. The URL in the address bar is https://portal.azure.com/#@durhamrick123@yahoo.onmicrosoft.com/resource/subscriptions/7f1957e6-ab9f-44e4-82eb-9e4b6cb8a45... . The current view is 'Resource groups > project15group1'. The left sidebar contains sections for Overview, Activity log, Access control (IAM), Tags, Events, Settings (Resource costs, Deployments, Security, Policies, Properties, Locks), Monitoring (Insights (preview), Alerts, Metrics, Diagnostic settings, Logs, Advisor recommendations, Workbooks), and a file download section at the bottom. The main content area displays the 'Essentials' tab for the resource group 'project15group1'. It shows the following details:

- Subscription (change) : Microsoft Azure Sponsorship 2
- Subscription ID : 7f1957e6-ab9f-44e4-82eb-9e4b6cb8a45...
- Tags (change) : Click here to add tags
- Deployments : 1 Failed, 5 Succeeded
- Location : East US

A table lists 21 records, showing columns for Name, Type, and Location. The data is as follows:

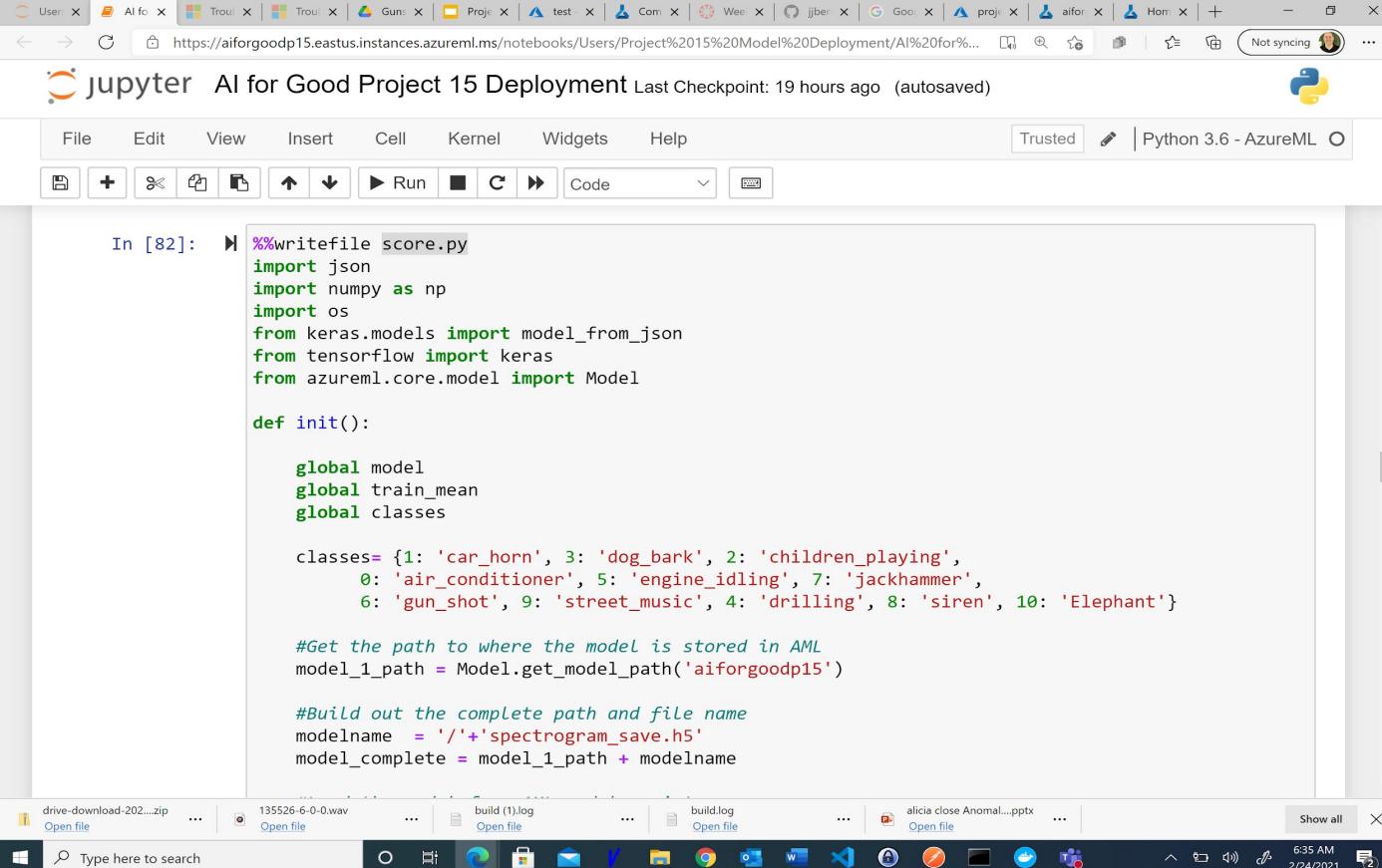
Name	Type	Location
aiforgoodgp1	Container registry	East US
aiforgoodgp1	Machine learning	East US
aiforgoodgp16071246797	Key vault	East US
aiforgoodgp18779954838	Application Insights	East US
aiforgoodgp19848574132	Storage account	East US
aiforgoodp15v3-cxYrYjV4E6BjexcGTjKxA	Container instances	East US
aiforgoodp15v4-cxYrYjV4E6BjexcGTjKxA	Container instances	East US
aiforgoodp15v5-cxYrYjV4E6BjexcGTjKxA	Container instances	East US
aiforgoodp15v6-cxYrYjV4E6BjexcGTjKxA	Container instances	East US
IoTHubSystemTopic	Event Grid System Topic	East US
P15-DPS-6smr6	Device Provisioning Service	East US

At the bottom, there are navigation links for < Previous, Page 1 of 1, and Next >, along with a 'Show all' button. The taskbar at the bottom shows several open files: drive-download-202....zip, 135526-6-0-0.wav, build(1).log, build.log, alicia close Anomal...pptx, and a file named 'Type here to search'.

# (Creation of Azure Machine Learning Environment and Registration of the model)

The screenshot shows the Microsoft Azure Machine Learning Studio interface. The browser address bar displays the URL: <https://ml.azure.com/model/aiforgoodp15:2/details?wsid=/subscriptions/7f1957e6-ab9f-44e4-82eb-9e4b6cb8a45e/resourcegroups...>. The main content area is titled "aiforgoodp15:2" and shows the "Artifacts" tab selected. The "output" folder contains two files: "file.npz" and "spectrogram\_save.h5". A decorative graphic of documents and a portrait photo is centered below the file list. The bottom navigation bar shows several open tabs, including "drive-download-202....zip", "135526-6-0-0.wav", "build (1).log", "build.log", "alicia close Anomal...pptx", and "Type here to search". The system tray at the bottom right indicates the date and time as 6:20 AM 2/24/2021.

# (Creation of Scory.py file needed for deployment)



In [82]:

```
%>>> %%writefile score.py
import json
import numpy as np
import os
from keras.models import model_from_json
from tensorflow import keras
from azureml.core.model import Model

def init():

    global model
    global train_mean
    global classes

    classes= {1: 'car_horn', 3: 'dog_bark', 2: 'children_playing',
              0: 'air_conditioner', 5: 'engine_idling', 7: 'jackhammer',
              6: 'gun_shot', 9: 'street_music', 4: 'drilling', 8: 'siren', 10: 'Elephant'}

    #Get the path to where the model is stored in AML
    model_1_path = Model.get_model_path('aiforgoodp15')

    #Build out the complete path and file name
    modelname  = '/'+'spectrogram_save.h5'
    model_complete = model_1_path + modelname
```

# (Create Scory.py model inference results test scoring with real gunshot)

The screenshot shows a Jupyter Notebook interface running on an AzureML instance. The notebook title is "jupyter AI for Good Project 15 Deployment". The code cell contains Python code for processing a spectrogram and predicting a class. The output cell (In [89]) shows the command run('...') and its result ([6], ['gun\_shot']). A red box highlights this output cell.

```
#Convert the spectrogram to a format for the model can use
test= np.array(glob(filename))
img_size=32
Input = np.stack([np.asarray(Image.open(l).convert('RGB')).resize((img_size, img_size))) for l in
##Predict the class using the model
output = model.predict((Input - train_mean[None,None,None,:])/256.)
preds = [np.argmax(l) for l in output]
cls_preds = [classes[l] for l in preds]

#Return the results of the prediction
return (preds,cls_preds)
```

In [80]: `init()`

In [89]: `run('./nn01a_20180203_1230-Copy1.wav')`

Out[89]: `([6], ['gun_shot'])`

In [82]: `%writefile score.py`  
`import json`  
`import numpy as np`

drive-download-202....zip ... 135526-6-0-0.wav ... build(1).log ... build.log ... alicia close Anomal....pptx ... Show all

# (Deployment of the Model to Azure ACI)

```
In [87]: inference_config = InferenceConfig(entry_script="score.py", environment=myenv)
aciconfig = AciWebbservice.deploy_configuration(cpu_cores=2,
                                                auth_enabled=True, # this flag generates API keys to secure access
                                                memory_gb=2,
                                                tags={'name': 'aiforgoodp15', 'framework': 'Keras'},
                                                description='Keras MLP on wav')

registered_model = Model(ws,'aiforgoodp15')

service = Model.deploy(workspace=ws,
                      name='aiforgoodp15v6',
                      models=[registered_model],
                      inference_config=inference_config,
                      deployment_config=aciconfig)

service.wait_for_deployment(True)
print(service.state)
```

Tips: You can try get\_logs(): <https://aka.ms/debugimage#dockerlog> or local deployment: <https://aka.ms/debugimage#debug-local> to debug if deployment takes longer than 10 minutes.

Running.....

Succeeded

ACI service creation operation finished, operation "Succeeded"

Healthy

Tip: If something goes wrong with the deployment, the first thing to look at is the logs from the service by running the following command:

```
In [ ]: print(service.get_logs())
```

This is the scoring web service endpoint:

drive-download-202....zip ... 135526-6-0-0.wav ... build (1).log ... build.log ... alicia close Anomal....pptx ... Show all

Type here to search

# (Deployment of the Model to Azure ACI)

The screenshot shows the Microsoft Azure Machine Learning portal interface. The main title bar reads "Microsoft Azure Machine Learning". The left sidebar has a vertical list of icons: Home, Create, Data, Compute, Models, Metrics, Datasets, Pipelines, and Monitoring. The "Endpoints" icon is selected. The main content area shows a list of endpoints under "aiforgoodgp1" with "aiforgoodp15v6" selected. The endpoint details are displayed:

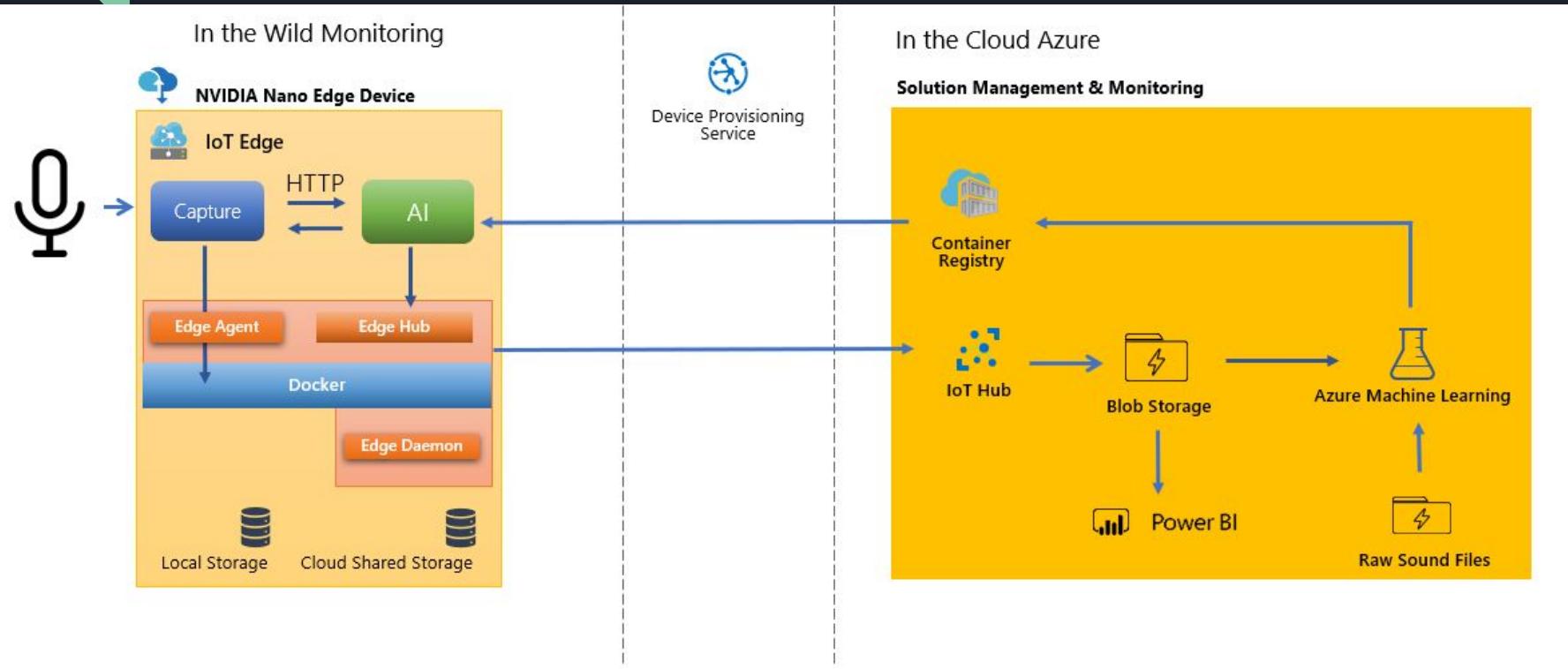
- Details**: Shows "Rick Durham" as the owner, "Model ID: aiforgoodp15v6", "Created on: 2/23/2021 6:26:01 PM", and "Last updated on: 2/23/2021 6:26:01 PM".
- REST endpoint**: The URL <http://6c9792fc-56c3-4631-8fa1-35013661721e.eastus.azurecontainer.io/score> is highlighted with a red border.
- Key-based authentication enabled**: This section is also highlighted with a red border.
- Swagger URI**: Shows "--".
- CPU**: Shows "2".
- Memory**: Shows "2 GB".
- Application Insights enabled**: Shows "false".

At the bottom of the screen, the taskbar shows several open files: "drive-download-202....zip", "135526-6-0-0.wav", "build (1).log", "build.log", "alicia close Anomal...pptx", and a search bar with "Type here to search". The system tray shows the date and time as "6:39 AM 2/24/2021".

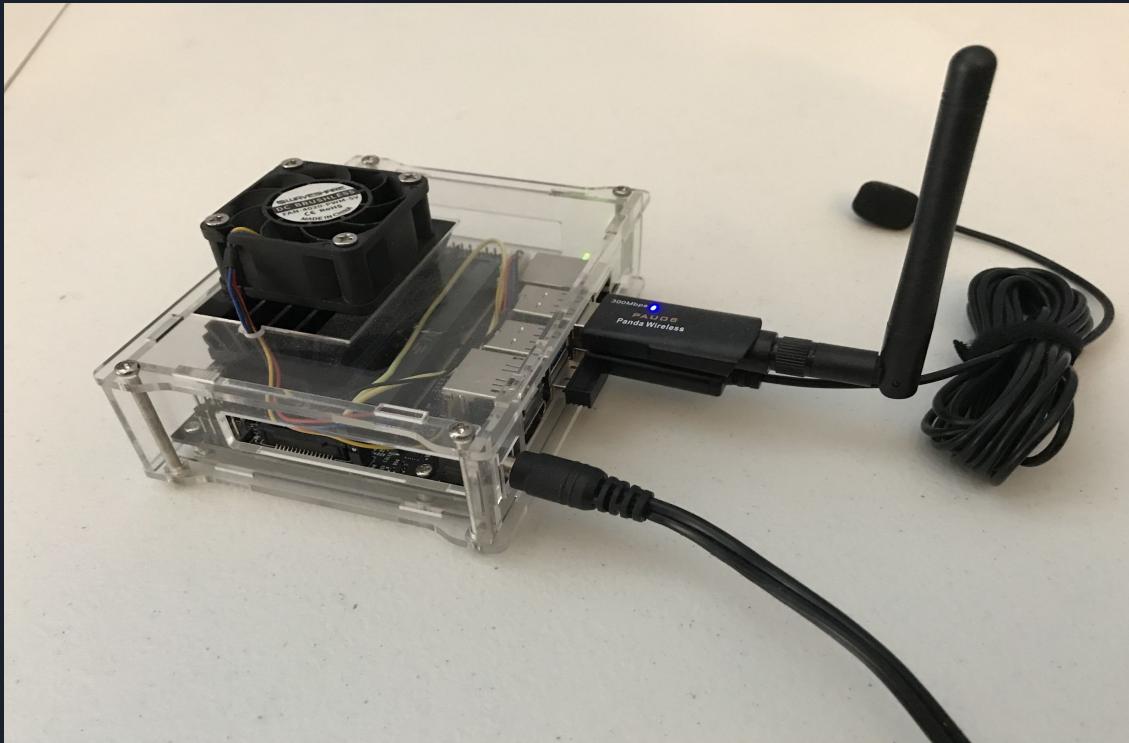


Deploy preparation of IoT Edge Device

# Proposed Cloud/Edge Design Architecture



The following device was created for Project 15  
based on the proposed architecture using a NVIDIA (Nano)



The following VSC Deployment Environment code was prepared for the - NVIDIA (Nano)

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS".
  - `deployment.template.json`
  - `INTELLIGENTEDGEHOL-MASTER` folder containing:
    - `.vscode`
    - `config`
    - `docker` folder with:
      - `jetson-nano-l4t`
      - `jetson-nano-l4t-cuda`
      - `jetson-nano-l4t-cuda-cudnn`
      - `jetson-nano-l4t-cuda-cudnn-opencv`
      - `jetson-nano-l4t-cuda-cudnn-opencv-darknet`
    - `modules/YoloModule` folder with:
      - `app`
      - `build`
      - `.gitignore`
      - `Dockerfile.arm64v8`
      - `module.json`
      - `.env`
      - `.gitignore`
      - `CODE_OF_CONDUCT.md`
    - `deployment.template.json`
    - `LICENSE`
    - `README.md`
  - Terminal:** Shows build information:

```
Go version: go1.13.15
Git commit: 2291f61
Built: Mon Dec 28 16:14:16 2020
OS/Arch: windows/amd64
Context: default
Experimental: true
```
  - Output:** Shows the output of the build process.
  - Status Bar:** Shows the current file is `deployment.template.json`, the path is `IntelligentEdgeHOL-master`, and the status bar includes "Azure IoT Edge".



# Conclusion and next steps...

## Future

- Gather more data to support improved models in terms of accuracy and samples that more represent the environment where elephants live and are poached
- Complete testing of ACI model deployment using REST calls
- Download deployment Docker container on Edge device and test using real gunshot and elephant sounds
- Classify elephant sounds based on demographics
- Combine gunshots with ambient sounds (Fourier transform)
- Eventually rework the way we inference so that a spectrogram file does not have to be created and ingested by the Keras model. This would be more of an in memory operation.



# Reference Documentation

## Literature Review:

The following documents were reviewed and respective repos used on the project:

[Project 15 from Microsoft | project15](#)

[2004.07948.pdf \(arxiv.org\)](#)

[Urban Sound Classification – Part 1: sound wave, digital audio signal | by Ricky Kim | Towards Data Science](#)[Urban Sound](#)

[Urban Sound Classification – Part 2: sample rate conversion, Librosa | by Ricky Kim | Towards Data Science](#)

[Classification using Neural Networks | by Shubham Gupta | Towards Data Science](#)

Our Code: <https://github.com/Oxford-AI-Edge-2020-2021-Group-1/AI4GoodP15>