CEC

PROGRAMMING

CABOT TRAIL TEAM

CONTENT

01
OUR TEAM

OUR TEAM

Vinuyan Sivakolunthu

2nd Year SOEN

Joshua Vilda

2nd Year SOEN

Camille Granade

5th Year COEN

Jack Spiratos

4th Year SOEN

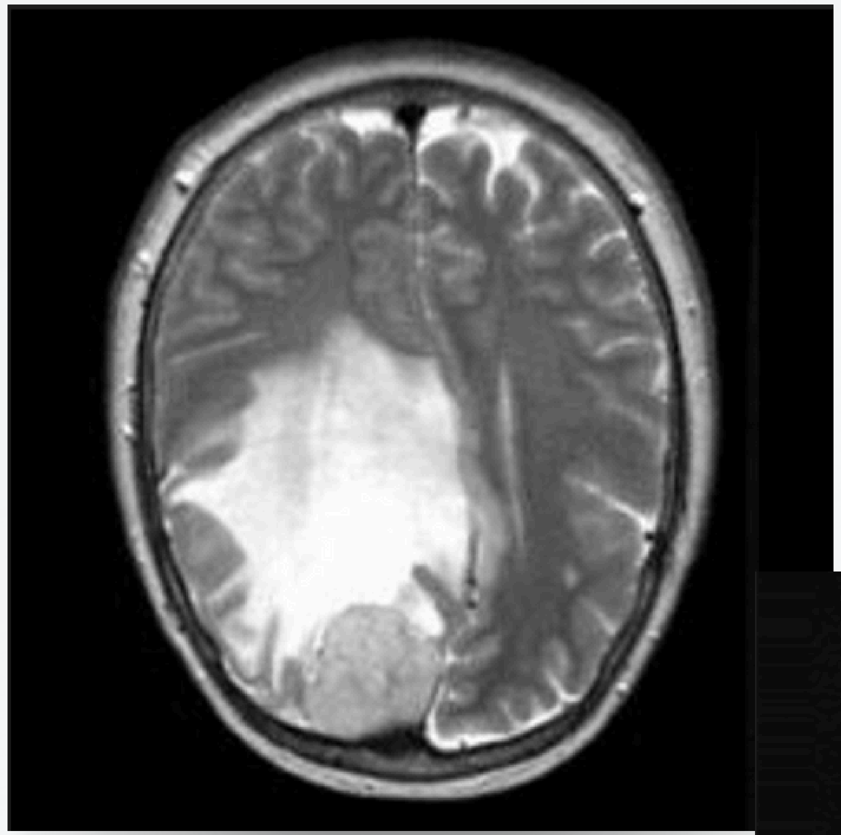
02
PROBLEM AND ETHICS

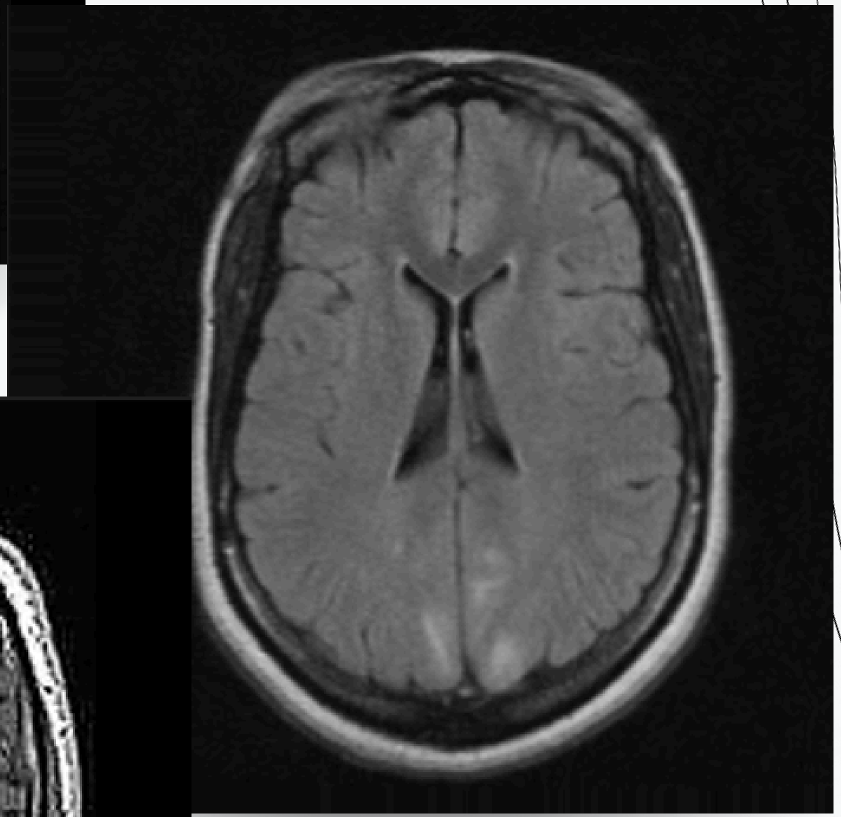
PROBLEM

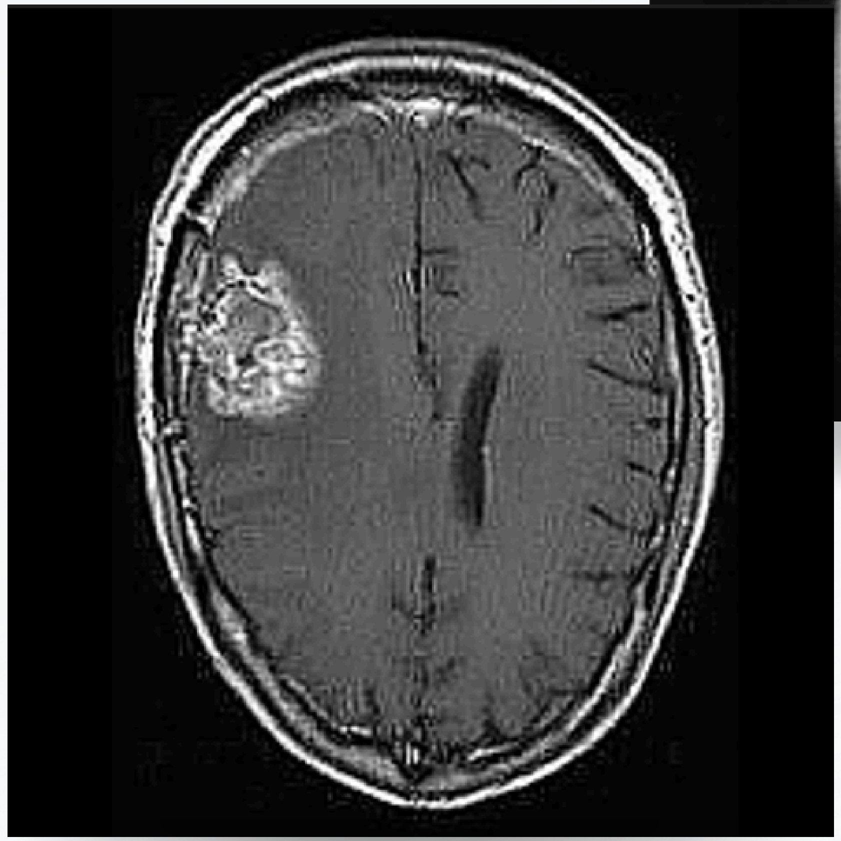
- Problematic wait times and inefficient resource use in hospitals

- JBOW needs program to aid in diagnosis and treatment of brain tumors

- Must consider ethical and liability concerns







OBJECTIVES

Valid

Check for the right item (brain tumor)

State-of-the-art

Is right a majority of the time
Trying to achieve >95%

Speed

limited by the competitions length - so speed is key

ETHICS & CONSIDERATIONS

- Bias in training data
  - Unknown Age/Gender/Background
  - Medical Bias
- Patient Safety and Accuracy First
  - False positives and negatives could create serious harm
- Explainability and Transparency
  - Medical staff should understand how the model arrives at its result
- Continous Validation and Improvement
  - Learn from new data, and catch model drift

03
INITIAL OPTIONS

CONSIDERED MODELS

Send the images with prompts to available LLM
Standard pre-trained language models are not trained for our use case

LANGUAGE MODEL API

You Only Look Once
Deep learning model for real-time visual object detection

YOLO

Using TensorFlow to train ResNet50 model,
A deep convolutional neural network architecture

TENSORFLOW RESNET-50

Using TensorFlow to train MobileNetV2 model,
A fast convolutional neural network architecture

TENSORFLOW MOBILENETV2

SECONDARY OPTIONS

Using pretrained models
Allows for quick and easy model use

PRETRAINED OUT-OF-THE-BOX

Training our own models
Takes longer and needs more resources
More accurate and more adapted to our use case

RETRAINING

DECISION MATRIX

Parameters	Categories of Evaluation					
Criteria	Ingenuity	Feasability	Accessibility	Reliability	Speed	
Parameters	Weighing of Categories					
Coarse Weight	100					
Fine Weight	15	30	15	30	10	
percentage	0.15	0.30	0.15	0.30	0.10	
Option description	Scoring					
API to Language Model	2	25	15	0	5	
TensorFlow + MoblieNetV2	13	24	13	25	7	
TensorFlow + Resnet-50	13	21	12	23	7	
YOLO	11	20	12	22	6	
Weighed Scores	Weighed Scoring					SCORE
API to Language Model	0.30	7.50	2.25	0.00	0.50	10.55
TensorFlow + MoblieNetV2	1.95	7.20	1.95	7.50	0.70	19.30
TensorFlow + Resnet-50	1.95	6.30	1.80	6.90	0.70	17.65
YOLO	1.65	6.00	1.80	6.60	0.60	16.65

04
FINAL MODEL

TENSORFLOW

an open-sourced machine learning framework

Gives access to the pretrained model we want

MOBILENET V2

Does not require the images to be labed as it assigns labels based on folder names

YES / NO

Allows us to use the GPU acceleration feature - allows for better and faster training

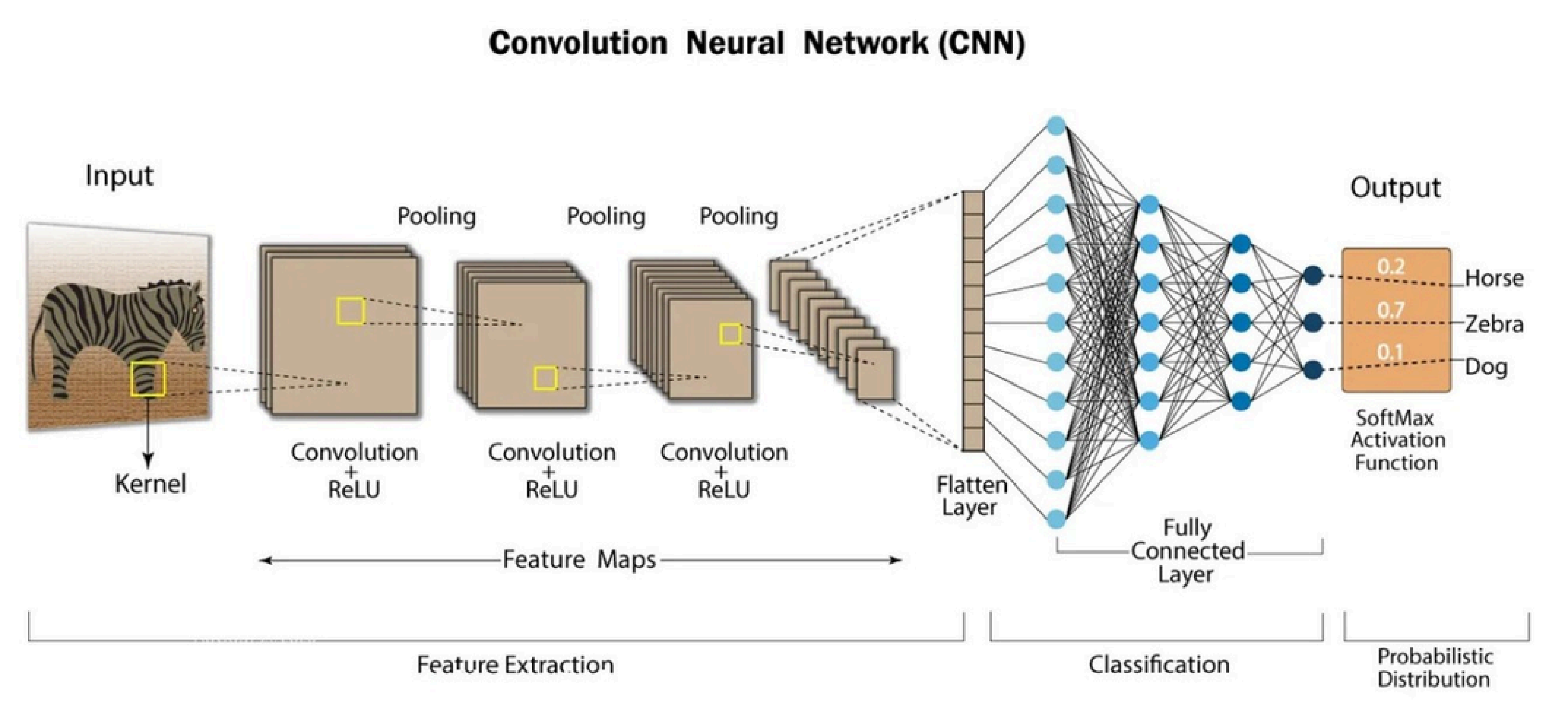
GPU

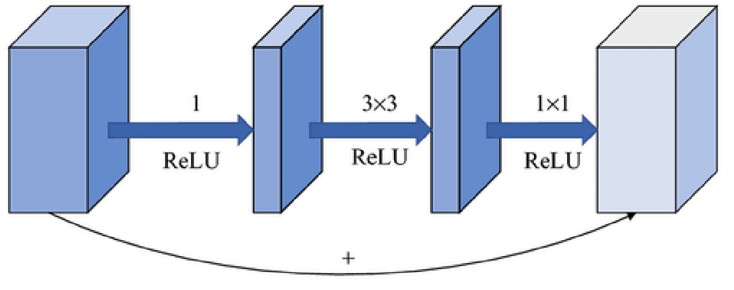
MOBILE NET V2

We wanted to use a CNN model for the classification of the images as they can directly process the images

We can train them such that:

- Early layers detect simple edges
- Middle layers detect textures & shapes
- Deep layers detect complex objects like our tumor structures





(a) Residual block

(b) Inverted residual block

INVERTED RESIDUALS

Avoids Vanashing Gradient problem by expanding input before depthwise convolution

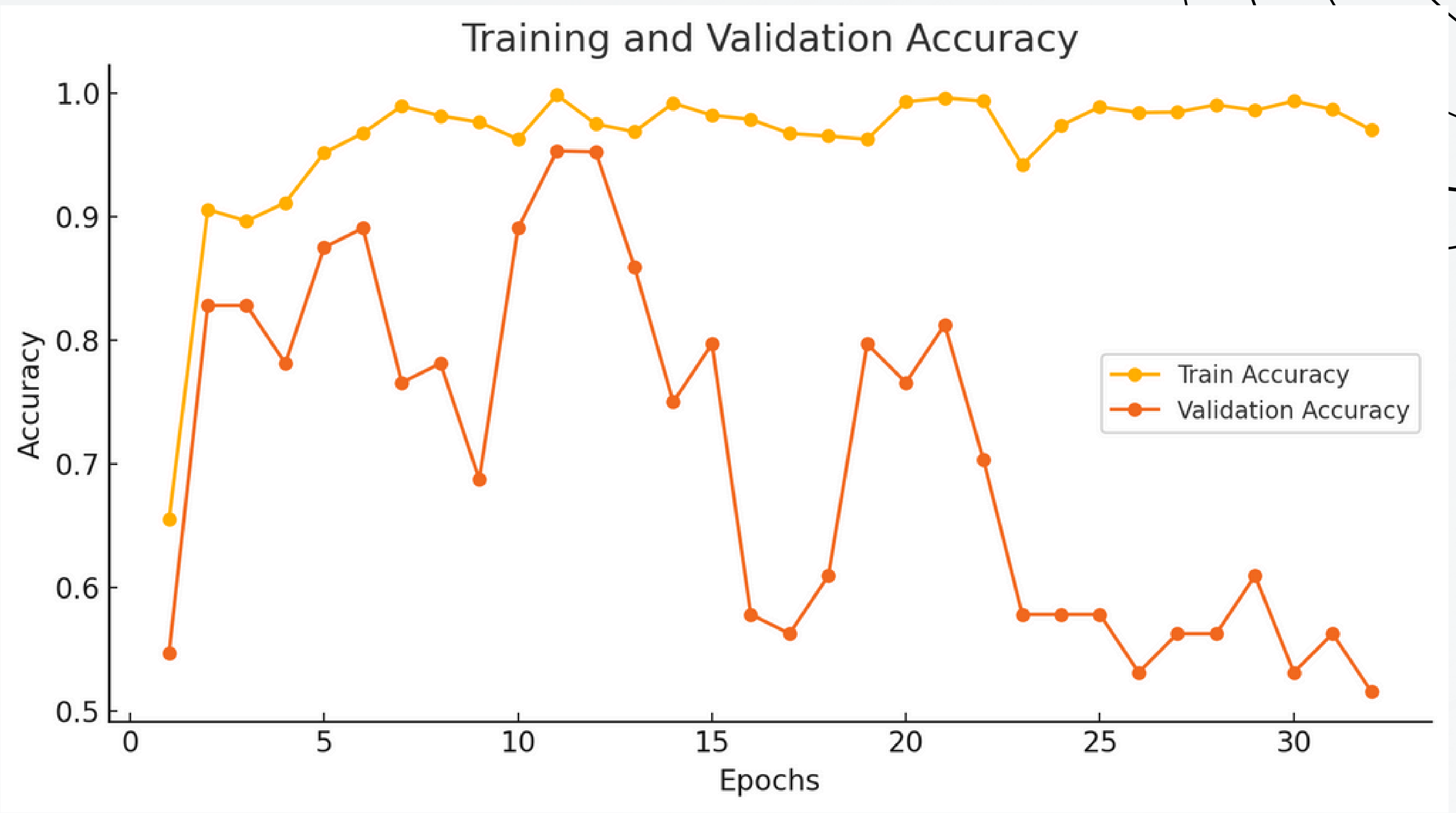
WHY MOBILE NET?

Provided good balance in terms of accuracy and time based on the size and variance of our data

PRE-TRAINED MODEL

It is faster, more accurate and requires less data
Trained on millions of data already

EPOCHS



Training and Validation Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
Accuracy
0
5
10
15
20
25
30
Epochs
Train Accuracy
Validation Accuracy

Training and Validation Loss
Train Loss
Validation Loss
20
15
10
5
0
Loss
0
5
10
15
20
25
30
Epochs

CODE

```python
base_model: Any = MobileNetV2(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
# Load ResNet50 without the top layer
# base_model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))

# Build classification head
x: Any = layers.GlobalAveragePooling2D()(base_model.output)
x: Any = layers.Dense(128, activation="relu")(x)
output: Any = layers.Dense(2, activation="softmax")(x)

# Create and compile model
model: Any = models.Model(inputs=base_model.input, outputs=output)
model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])

# Set dataset parameters
AUTOTUNE: Any = tf.data.AUTOTUNE
img_size: tuple[Literal[224], Literal... = (224, 224)
batch_size = 30

# Load dataset (automatically batches data)
dataset: Any = tf.keras.preprocessing.image_dataset_from_directory(
    C_2025", image_size=img_size, batch_size=batch_size, shuffle=True, seed=42
```

RUNNING

```python
# Evaluate on test data (if available)
model = tf.keras.models.load_model("brain_mri_classifier.keras")
class_names = ["no", "yes"]
# Load and preprocess the image
img_path = "C:/Users/joshv/OneDrive/Pictures/Documents/CEC_2025/no/no__894.png"
img = image.load_img(img_path, target_size=(224, 224))  # Resize
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)  # Add batch dimension
img_array = img_array / 255.0  # Normalize (same preprocessing as training)

# Make prediction
predictions = model.predict(img_array)

# Get the class with the highest probability
predicted_class = class_names[np.argmax(predictions)]
confidence = np.max(predictions)

print(f"Predicted class: {predicted_class} with {confidence:.2f} confidence")
```

TRAINING

EXPORTING

```python
def csv_output(output_text):
    with open("cabot_trail_output.csv", 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerows(output_text)

def csv_array_appender(imageName,yesNo,error,array):
    error = error*100
    errorPercentage = str(error)+ "%"
    array.append([imageName,yesNo,errorPercentage])
```

05
DIFFICULTIES AND TIMELINE

DIFFICULTIES ENCOUNTERED

DATA TYPE

PNG format not a standard format for analysis
Needed to reformat in code

LABELLING

Labelling would have made training more efficient
But labelling large dataset extremely ineffecicient

HARDWARE

Had issues using GPU's
Had to adapt to CPU usage

LARGE DATA SET

Meant training took a long time
Needed to find ways to improve training speed

TIMELINE

9:00-11:30

- Research and identify models and constraints
- Setup and test Yolo model
- Realized labelling problem

11:30-13:00

- Development of TensorFlow and ResNet50 on small dataset
- Input and output setup

13:00-14:00

- Try using GPU clusters
  - Switched to MobileNet

14:00-17:30

- Trained MobileNet on large amount of CPU
- Adjusting training parameters
- Tested model

TASK DISTRIBUTION

Camille	Scripts, Presentation
Joshua	Exporting, Refactoring, Testing
Vinuyan	Research, MobileNet, ResNet, Training
Jack	Research, Yolo, Training, Data Analysis

06
RESULTS

RESULTS

- Accuracy:
  - How many are results are true overall
- Precision:
  - Out of all the predicted tumors, how many were actually tumors
- Recall:
  - Out of all actual tumors, how many were correctly identified?
- F1 Score:
  - General metric of the model's predictive skill

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

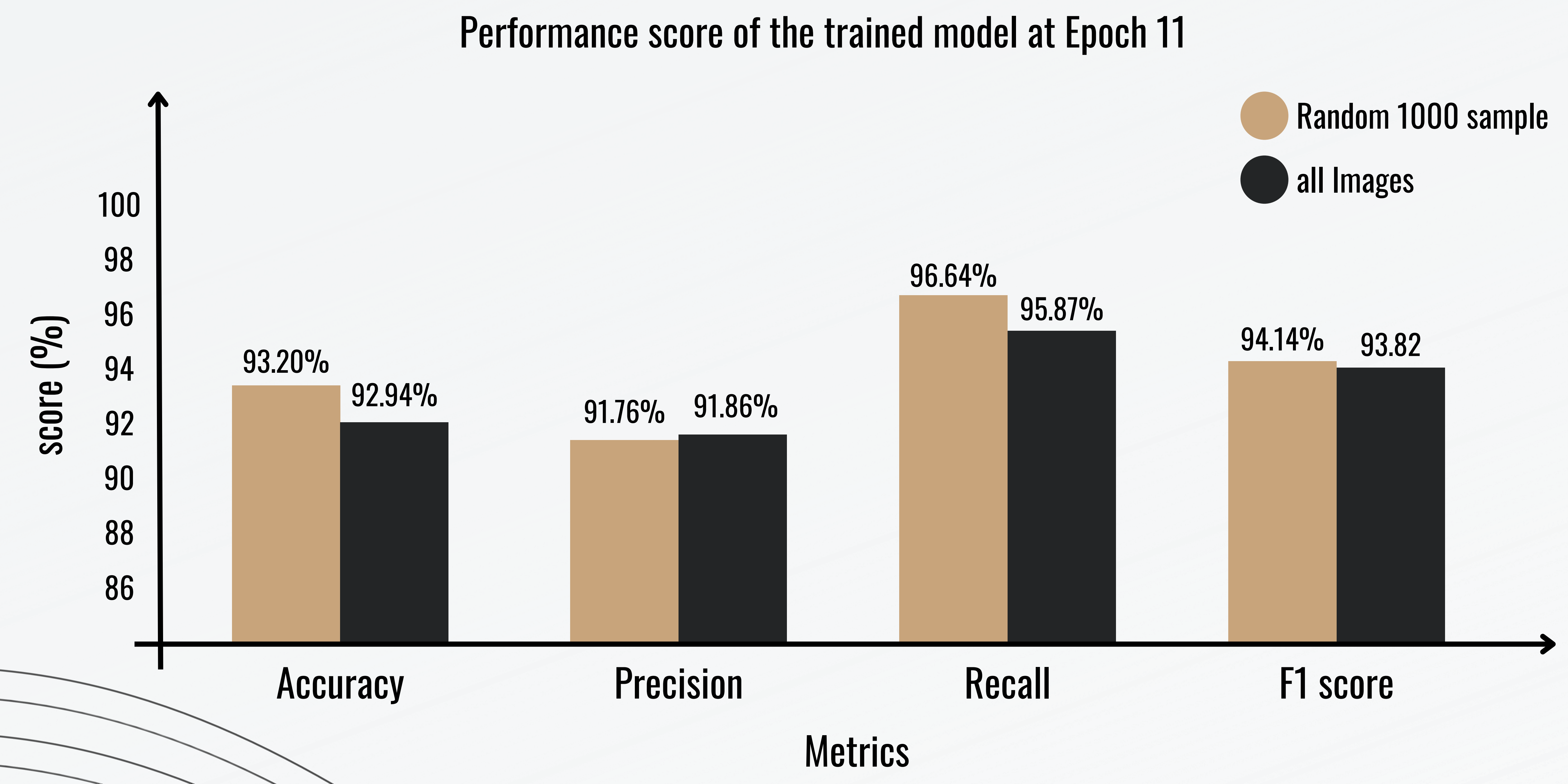
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

RESULTS



Performance score of the trained model at Epoch 11
Random 1000 sample
all Images
score (%)
100
98
96
94
92
90
88
86
93.20%
92.94%
91.76%
91.86%
96.64%
95.87%
94.14%
93.82
Accuracy
Precision
Recall
F1 score
Metrics

THANK YOU!

We'll gladly take your questions now

REFERENCES

[1] “ChatGPT,” Chatgpt.com, 2025. https://chatgpt.com/ (accessed Mar. 15, 2025).

[2] Ultralytics, “Python,” Ultralytics.com, 2023. https://docs.ultralytics.com/usage/python/#benchmark (accessed Mar. 15, 2025).

[3] J. B. Marion, J. A. Saez-Rodriguez, M. I. Jordan, and J. L. Reyes-Ortiz, "Evaluation Metrics for Machine Learning Models," arXiv preprint arXiv:1801.04381, 2018.