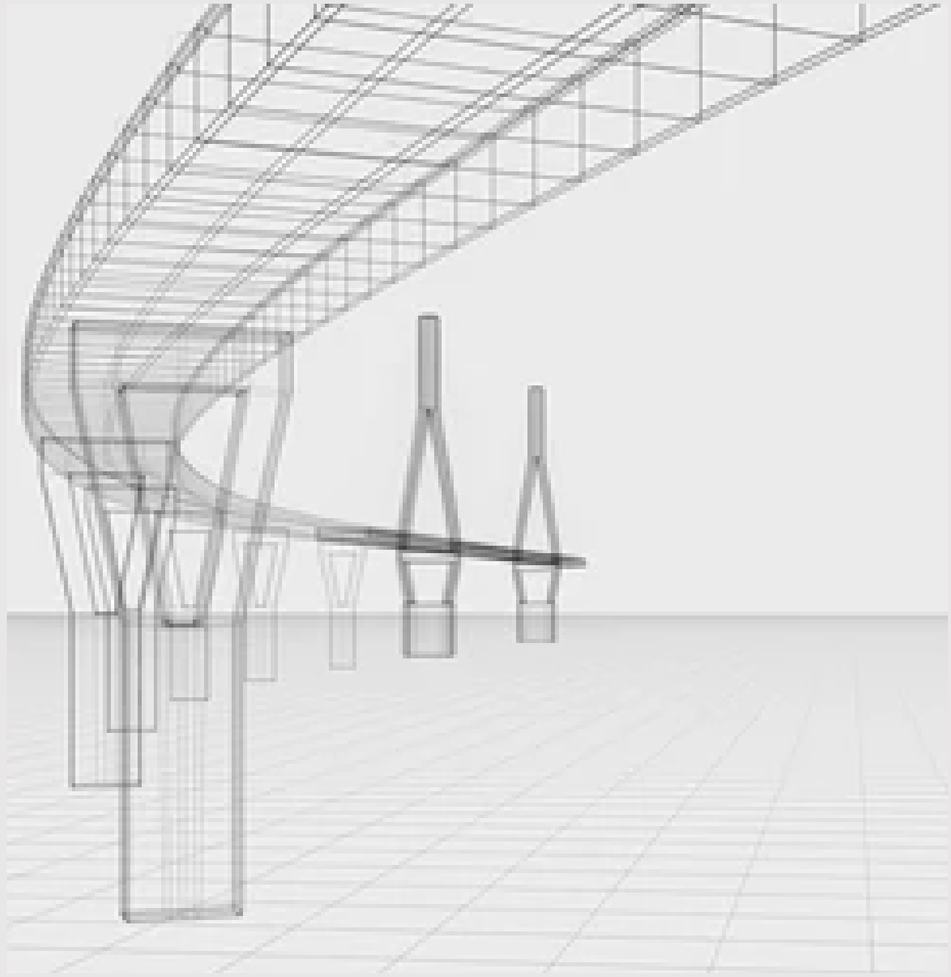

L&T EDUTECH HACKATHON
SHAASTRA IIT MADRAS 2023
BHARATFLY_CODERS
IT'S ALL ABOUT IMAGINEERING

About Us

The Bharatfly_Coders team, comprising of two members from the esteemed National Institute of Technology (NIT) Tiruchirapalli, participated in this highly competitive hackathon hosted by L&T and organized by the renowned Indian Institute of Technology Madras.

With a thorough analysis of the presented problem statement and a commitment to delivering the most optimal solution, we are excited to present the results of our diligent efforts. Let us begin the exploration.

PS1



Concrete cracking is a major issue in Bridge Engineering. Detection of cracks facilitates the design, construction and maintenance of bridges effectively. Design error, construction mistakes, hydraulic, collision and overload are the top 5 leading causes of bridge failures.

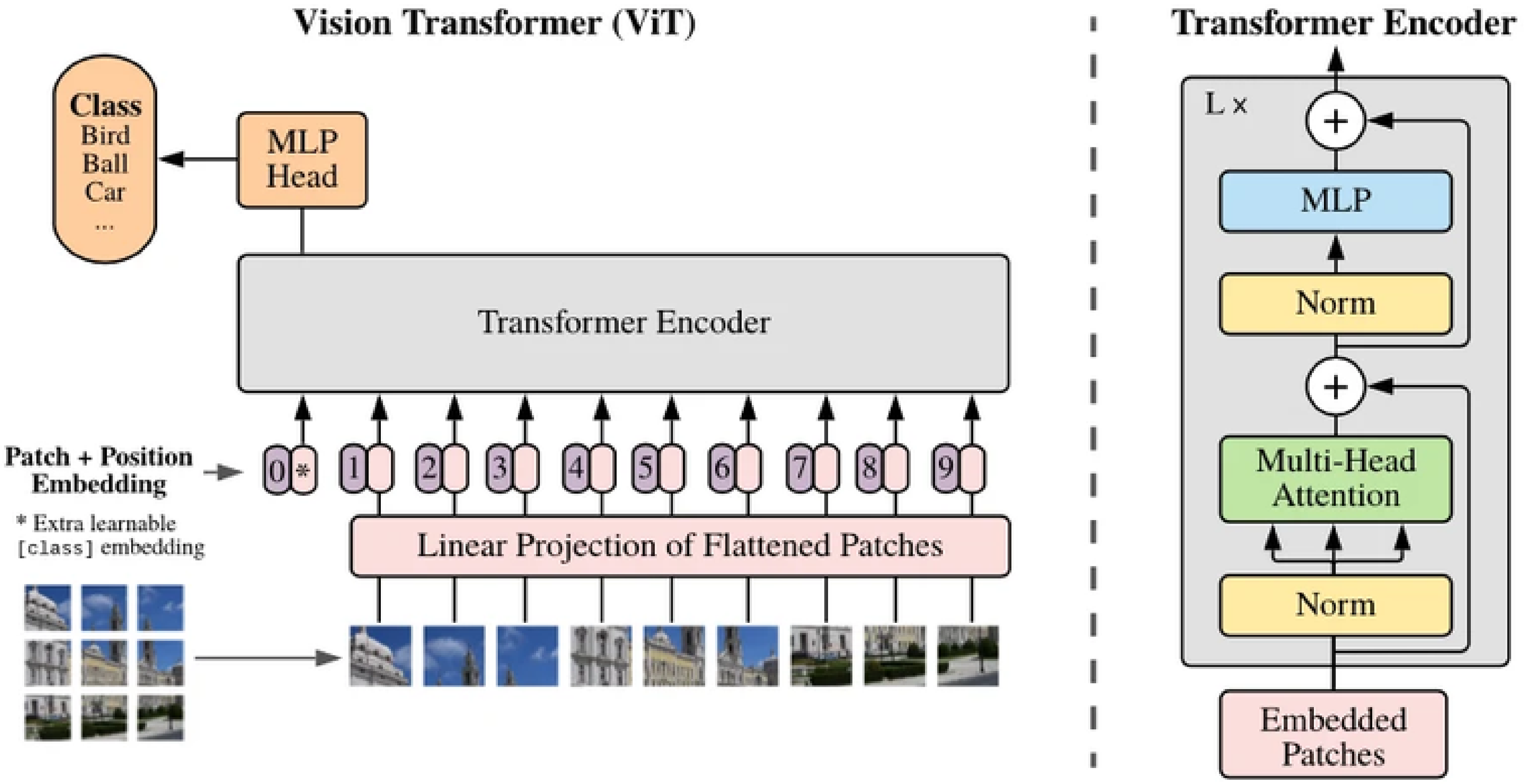
Given an open source dataset containing folders named positive and negative image set, develop a suitable Deep Learning framework which can detect the crack in the image from the dataset.

OUR APPROACH

- We started exploring the dataset and started fine-tuning the given dataset on the standard pre-trained model such as Resnet, VGG16, Efficientnet-B7, and Vision Transformers (ViT).

- After multiple runs, we found that ViT models outperformed all other models with/without augmentation and performed exceptionally well using the SGD optimizer in particular.

- Image augmentation, such as random rotation and flips, did not help in improving the model performance, and the model was able to generalize well without overfitting without any image augmentation. All the models were trained using Kaggle GPU and partially locally.

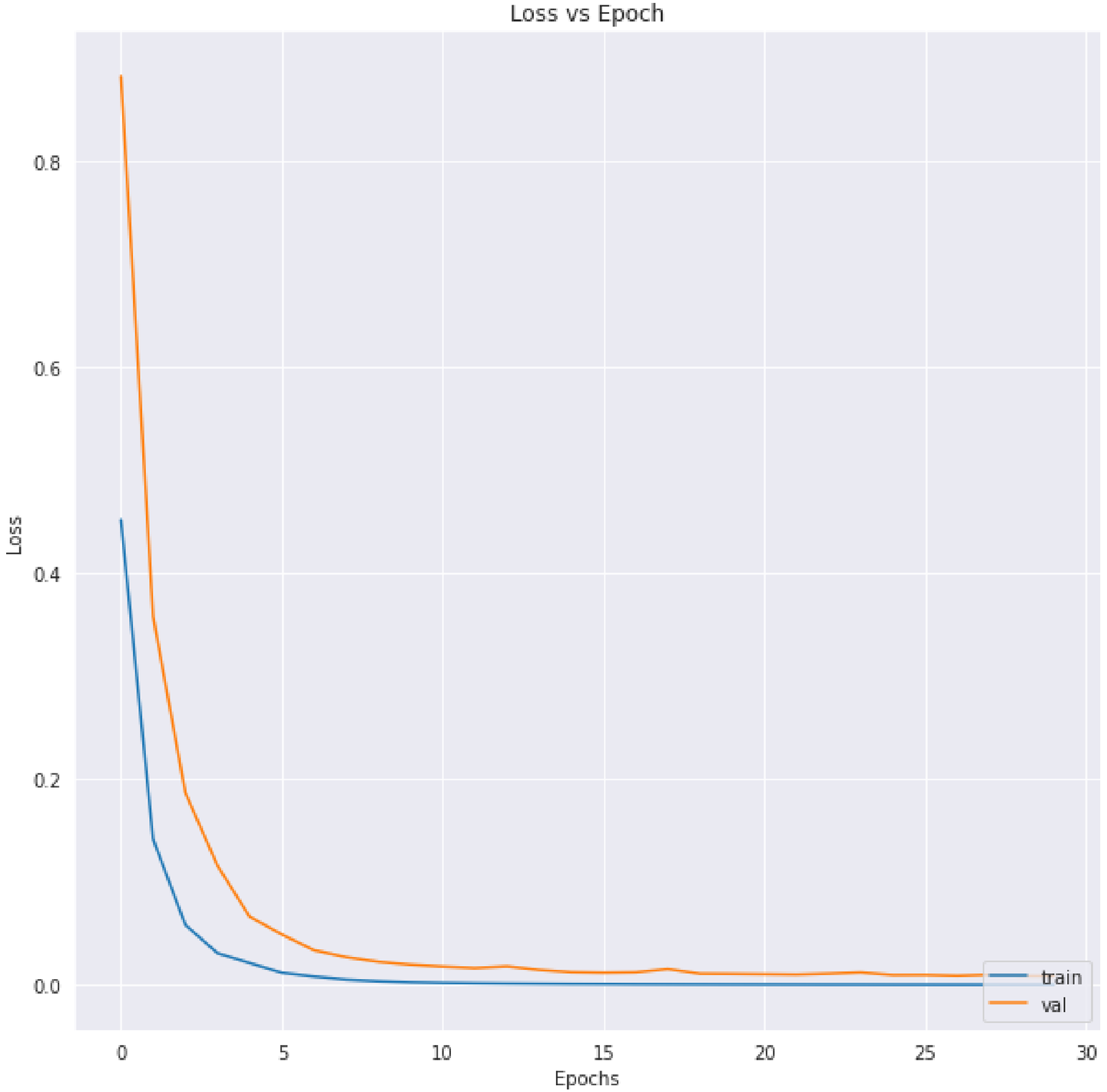

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP
Head
Transformer Encoder
Patch + Position
Embedding
* Extra learnable
[class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head
Attention
Norm
Embedded
Patches

OUR APPROACH

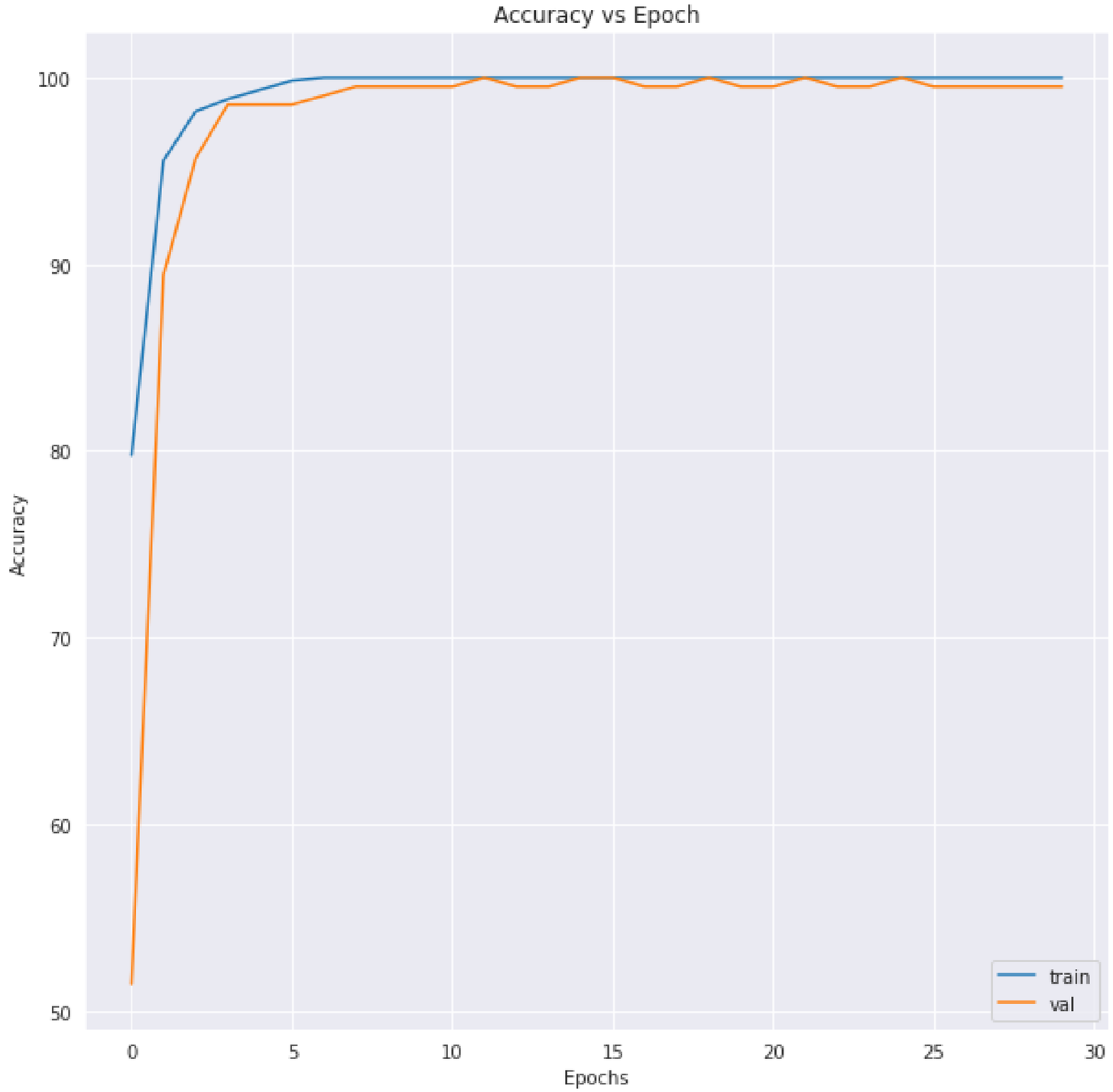
SNO	Model	Test Loss	Test Accuracy	Precision	Recall	F1 Score	Support	Size(mb)
1	vit_tiny_r_s16_p8_224	0.016840581	100	1	1	1	100	24.64
1a	vit_tiny_r_s16_p8_224(adam)	1.329885784	84.13461538	0.87878788	0.84	0.83579639	100	24.64
2	**vit_tiny_patch16_224**	**0.0086214**	**100**	**1**	**1**	**1**	**100**	**24.64**
3	vit_small_patch8_224_dino	0.001345925	100	1	1	1	100	86.74
4	vit_small_patch16_224	0.017205385	100	1	1	1	100	86.72
5	vit_base_patch8_224	0.029841557	99	0.99019608	0.99	0.989999	100	343.3
6	vit_base_patch16_224	0.000899294	100	1	1	1	100	343.26
7	vit_large_patch16_224	0.04213304	99.5	0.99504951	0.995	0.9949988	100	1.21gb

After testing out multiple versions of ViT pretrained models, we found out that **Vit_Tiny_Patch16_224** with SGD optimizer performed the best with test accuracy of 100, all other metrics such as Precision, Recall, F1 Score as 1 and relatively very small 24.64 mb.

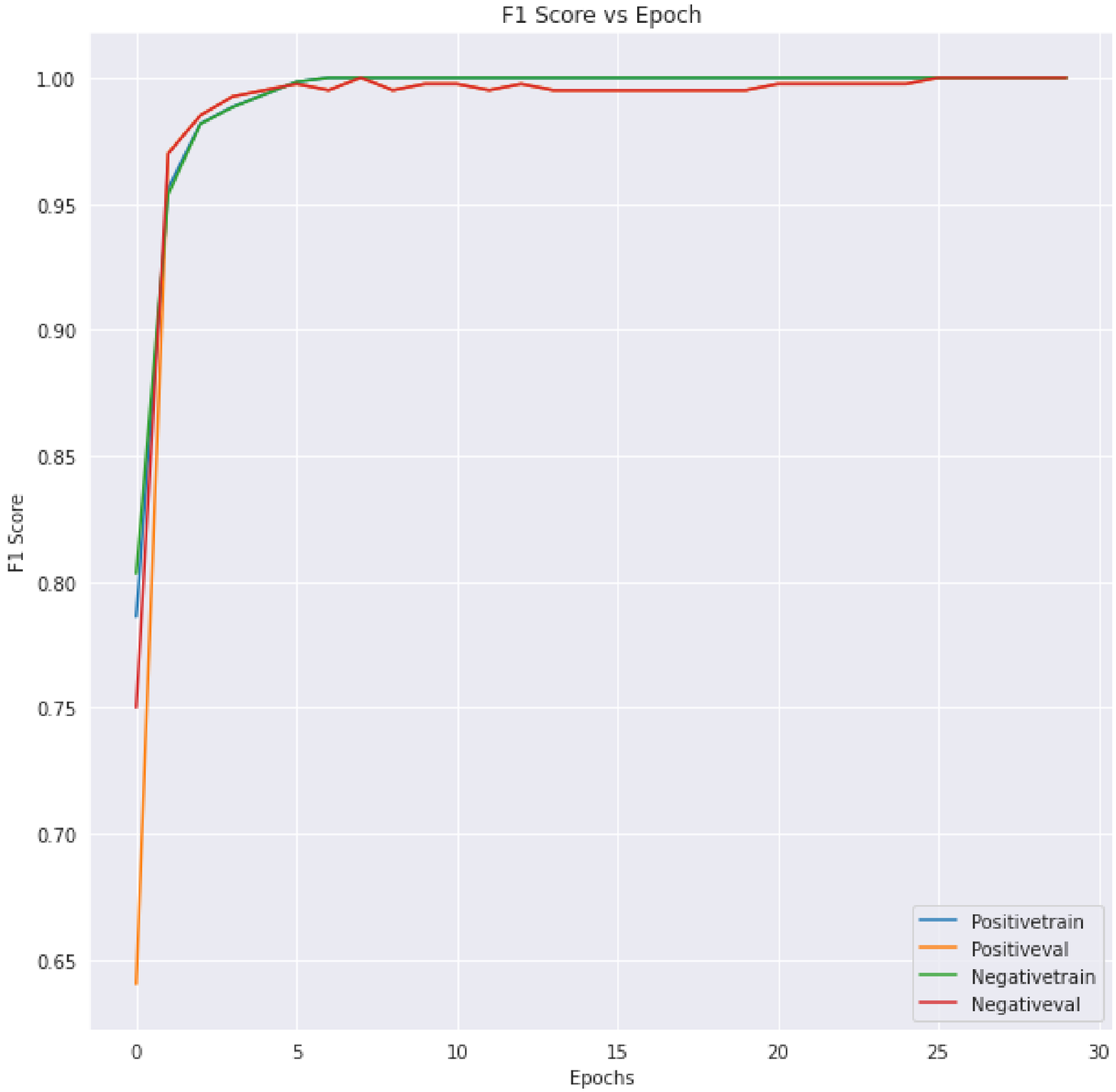
Loss vs Epoch



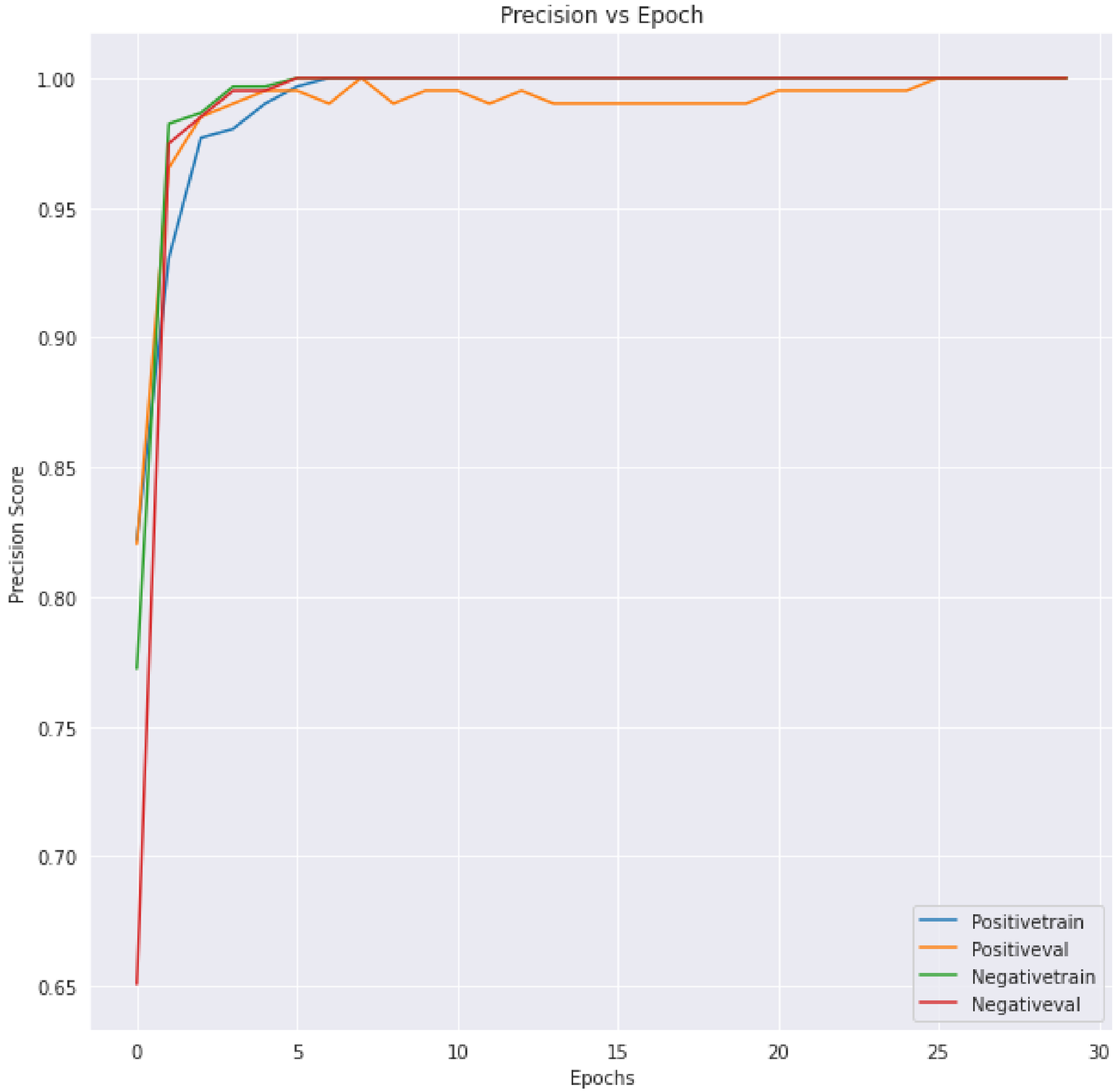
Accuracy vs Epoch



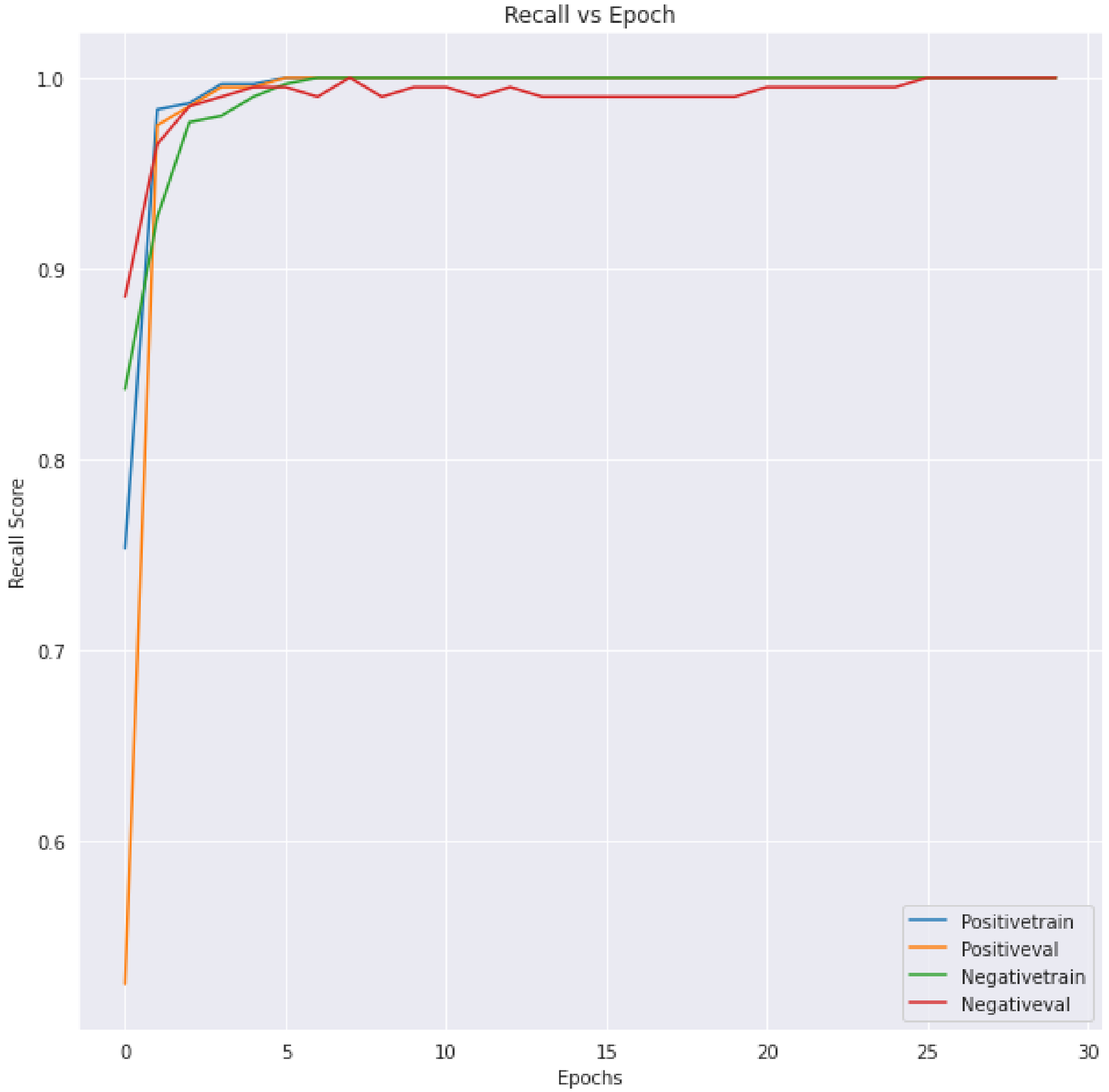
F1 Score vs Epoch



Precision vs Epoch



Recall vs Epoch



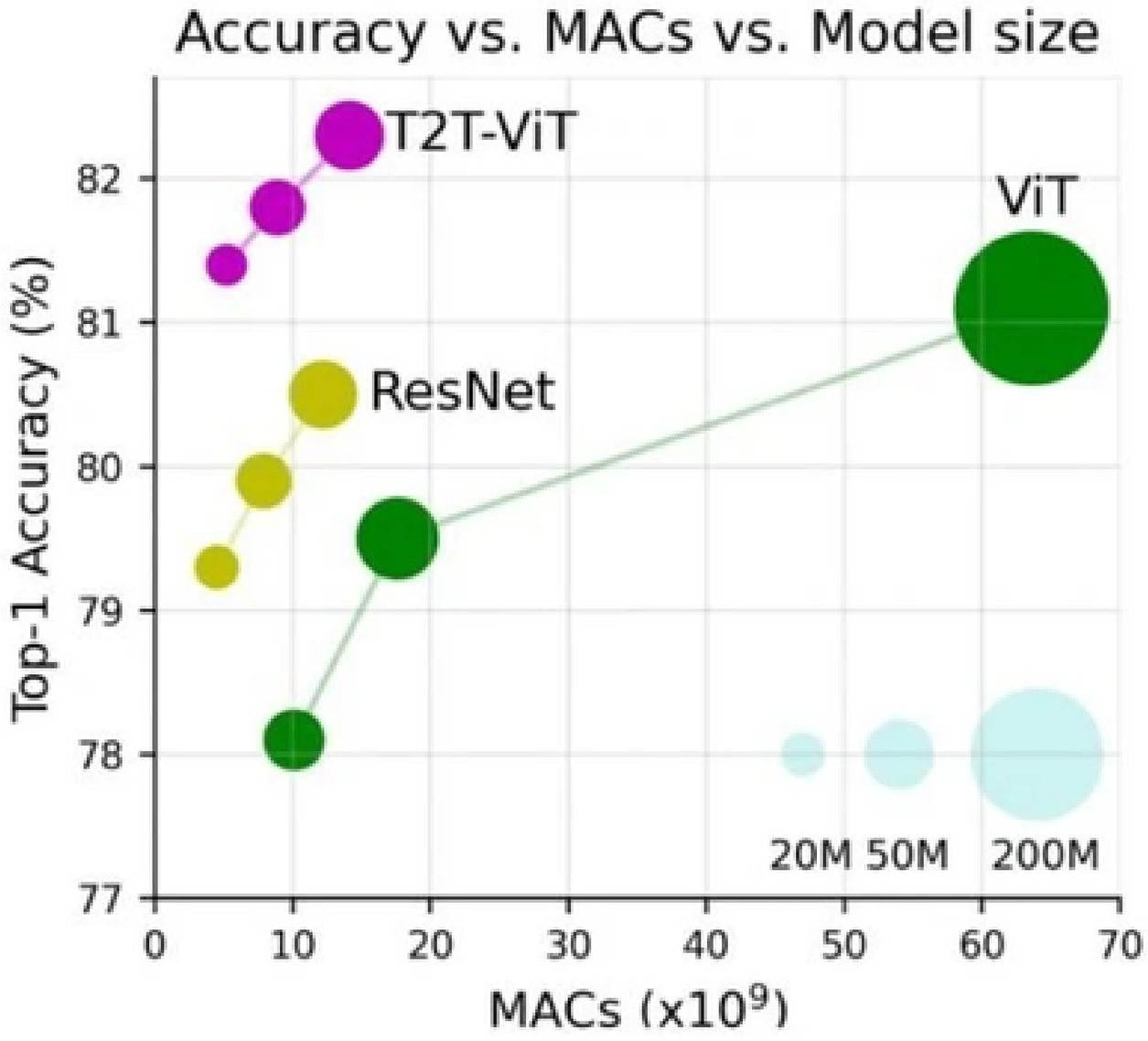
OUR APPROACH

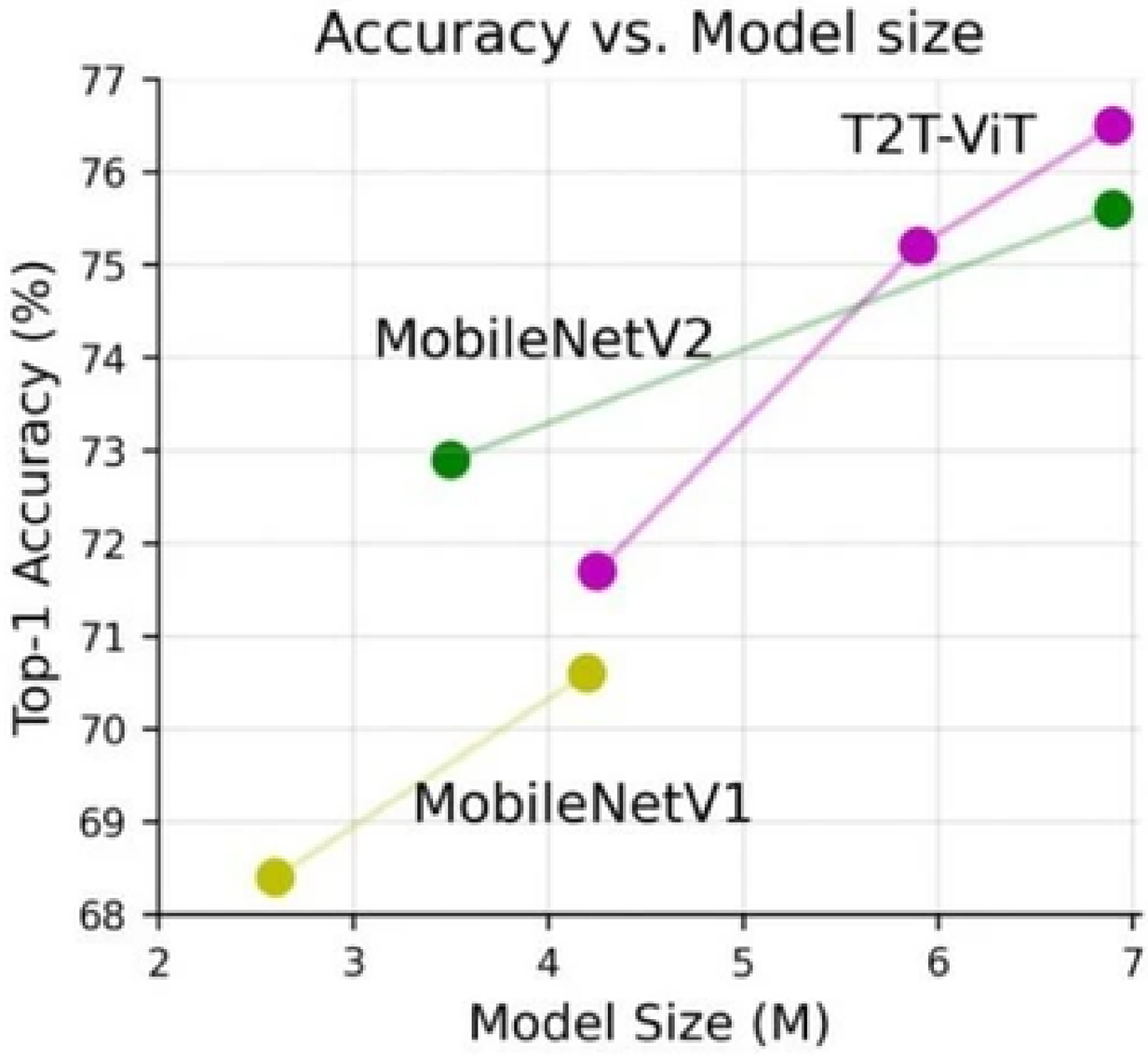
- While deploying the model in Pytorch Lite, the size is large with atleast 100-200 mb and is also not much developed in comparison with Tensorflow Lite in terms of production.

- So, we ran the Vision Transformer model with Tfimm in Tensorflow Lite which is the equivalent of Timm in Pytorch and is developed for deployment purposes which can be directly used at production level but the issue is Tfimm is not yet ready for production.

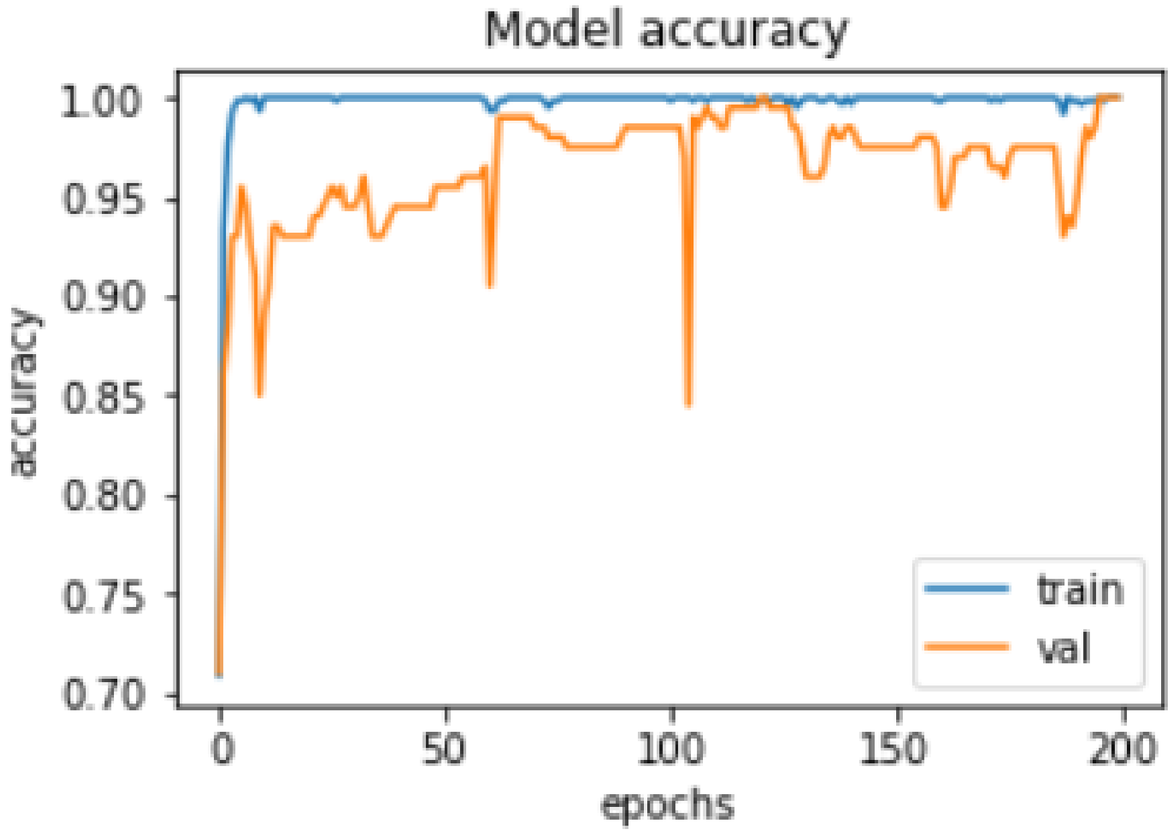
- After researching about various models, MobileNetV2 performed well in classification models with size of just 9.2 mb.

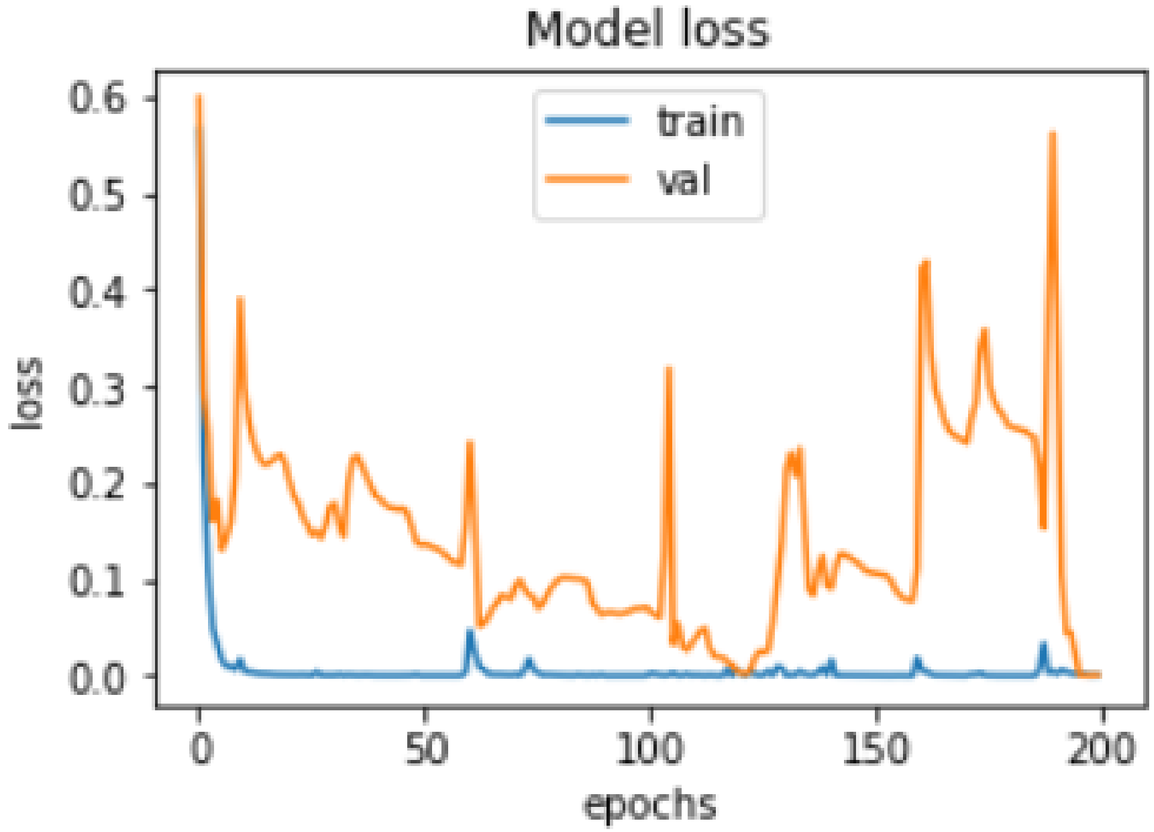
Network	Accuracy	Computation	Memory
MobileNetV2	81.44%	0.117 ms	8.9 MiB
ResNet50	79.41%	0.237 ms	90.2 MiB
DenseNet169	82.44%	0.367 ms	49.1 MiB
InceptionV3	83.59%	0.216 ms	83.6 MiB
InceptionResNetV2	83.13%	0.492 ms	208.4 MiB


Accuracy vs. MACs vs. Model size
T2T-ViT
ViT
ResNet
20M 50M 200M
Top-1 Accuracy (%)
MACs (x10^9)
82
81
80
79
78
77
0
10
20
30
40
50
60
70

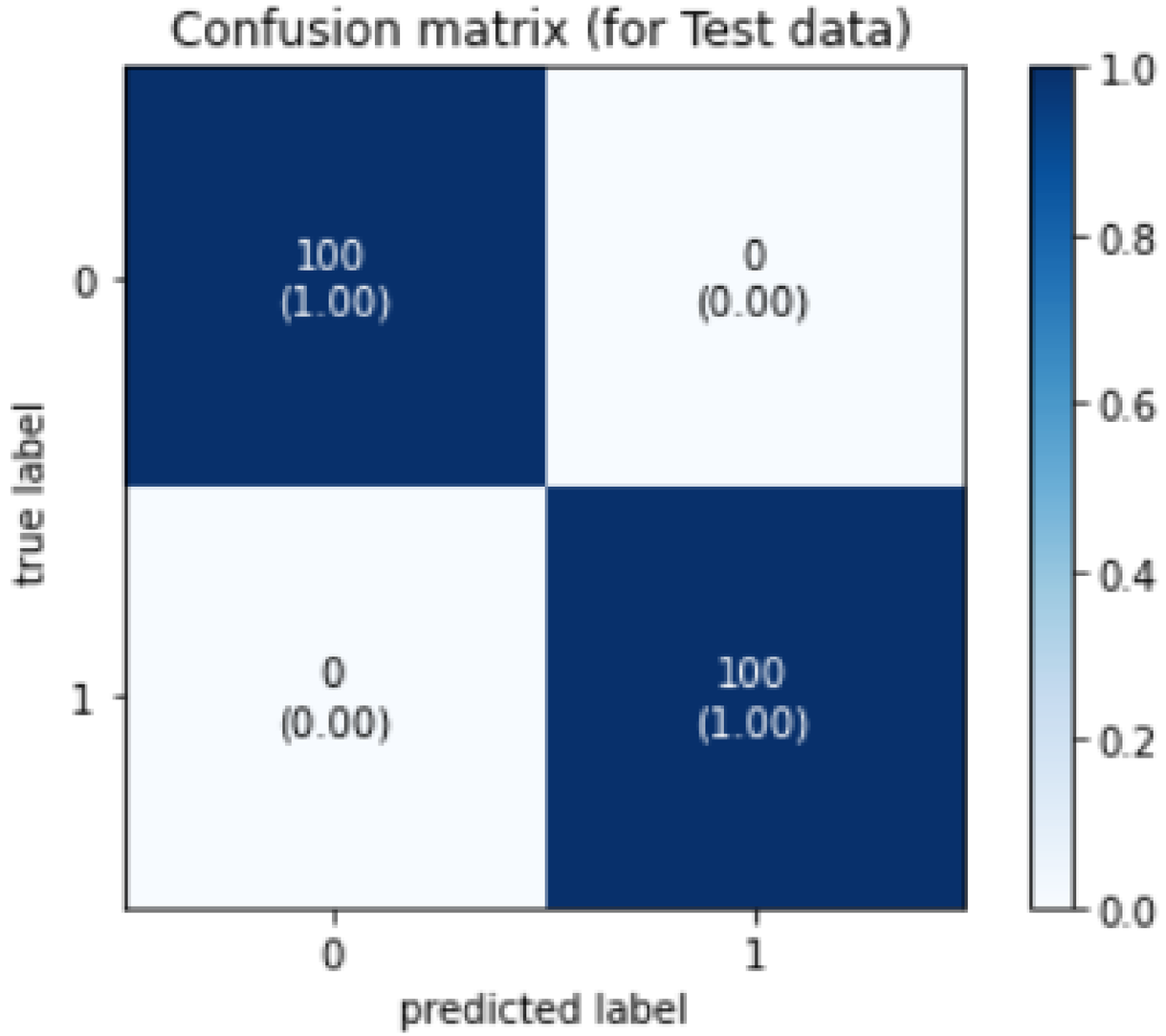

Accuracy vs. Model size
T2T-ViT
MobileNetV2
MobileNetV1
Top-1 Accuracy (%)
Model Size (M)
77
76
75
74
73
72
71
70
69
68
2
3
4
5
6
7

Accuracy vs Epoch and Loss vs Epoch

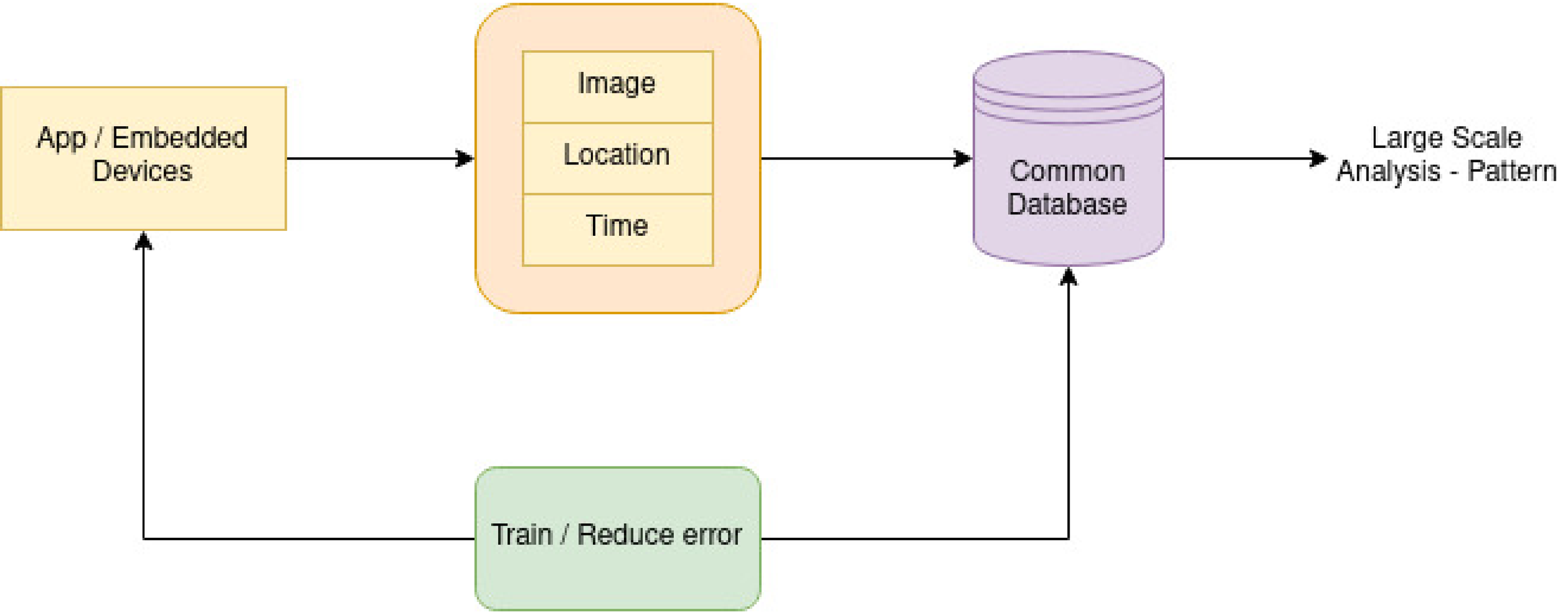




Confusion Matrix

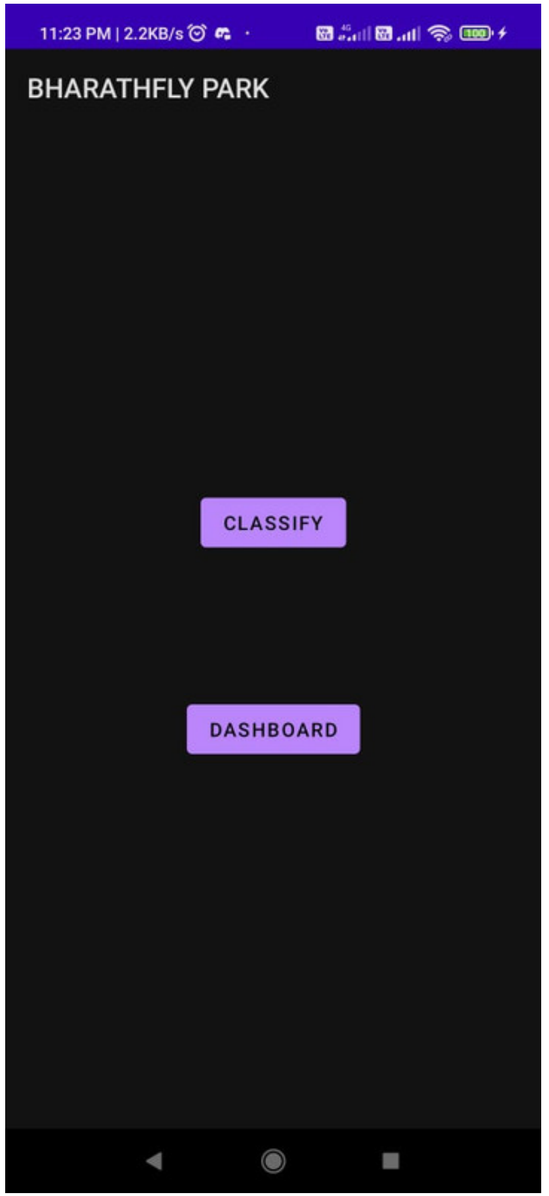


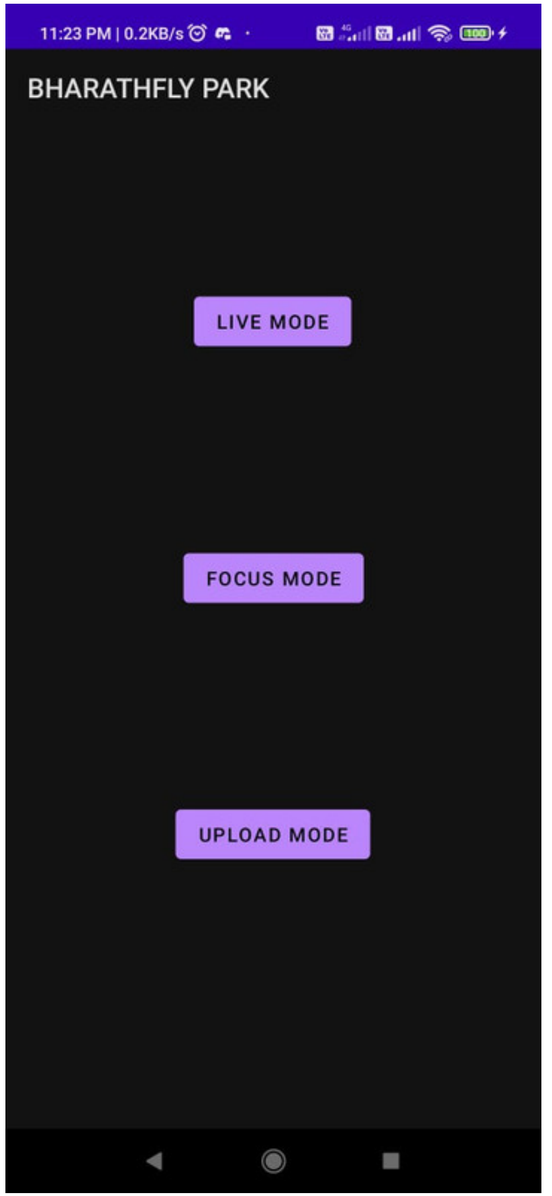
Design framework

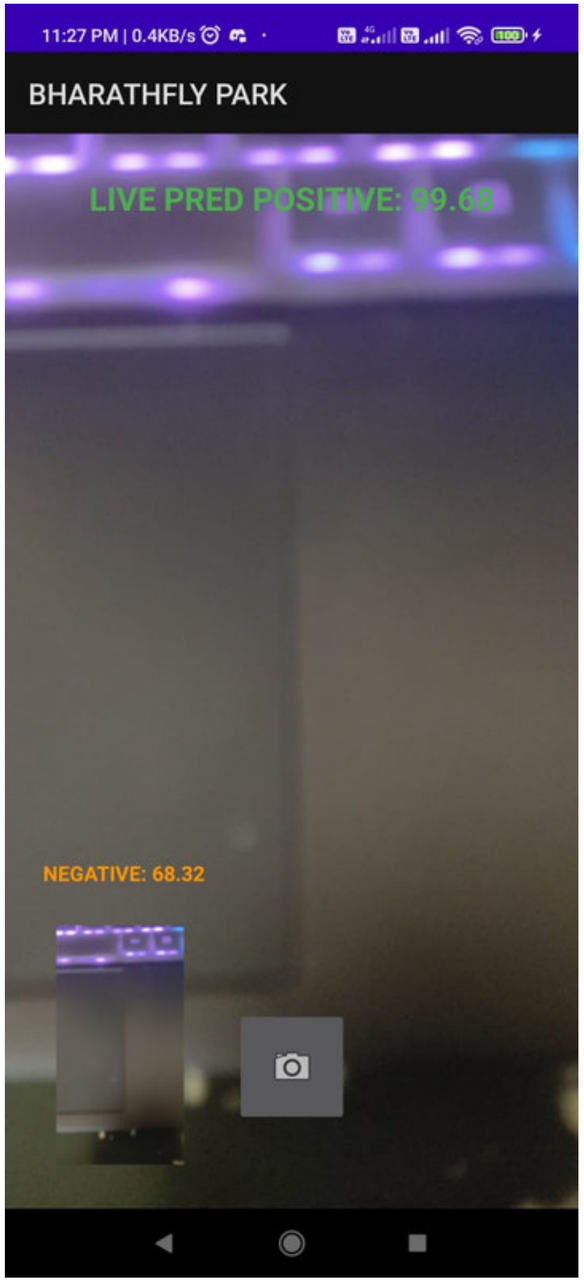


DEMO

- Three camera modes:
  - 1) Live - Classification on live cam feed
  - 2) Focus - Classifying a high-quality image at a time.
  - 3) Upload - Classify Images locally available on the phone.

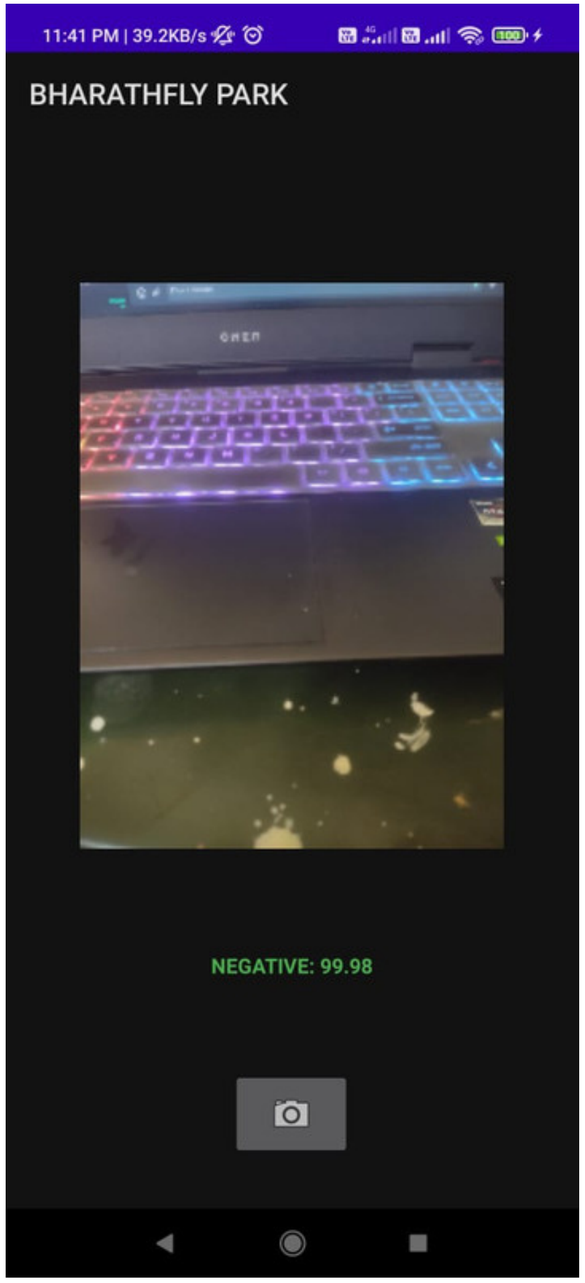






LIVE MODE

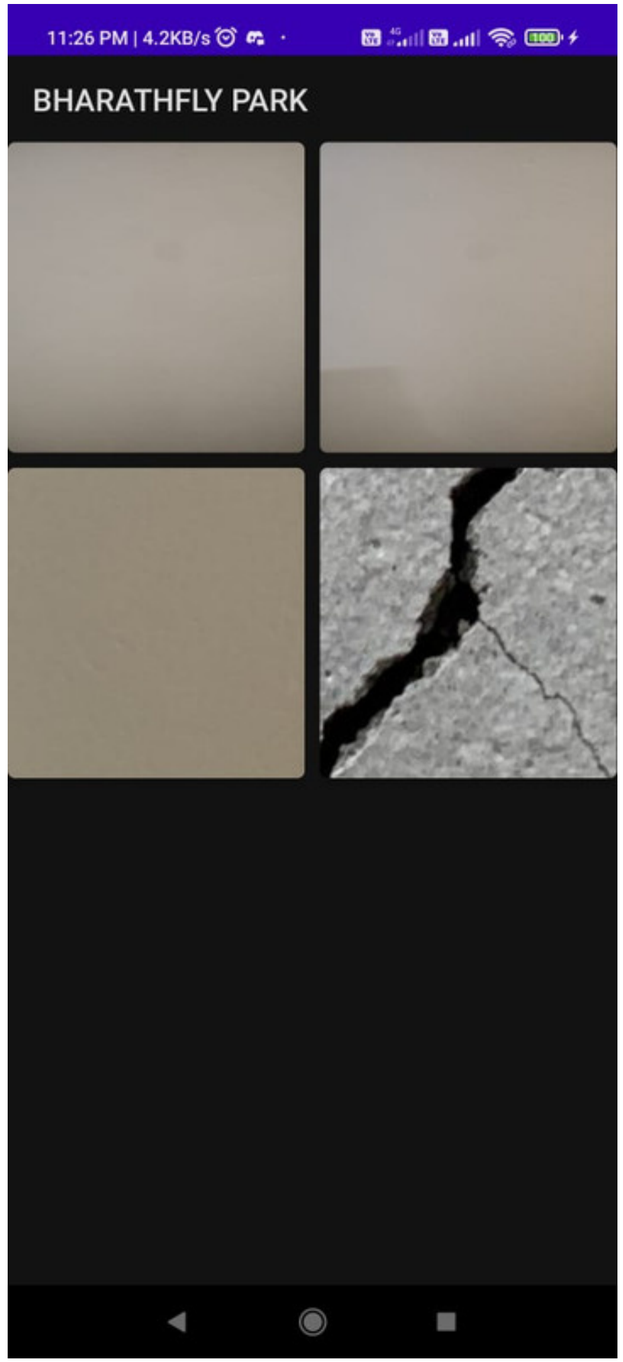
UPLOAD MODE



FOCUS MODE

DEMO

- Dashboard:
  - 1) Single View Model
  - 2) Gallery Mode
- Shows:
  - Image taken.
  - Predicted/Classified output.
  - Location at which the photo was taken.
  - Time at which it was taken.



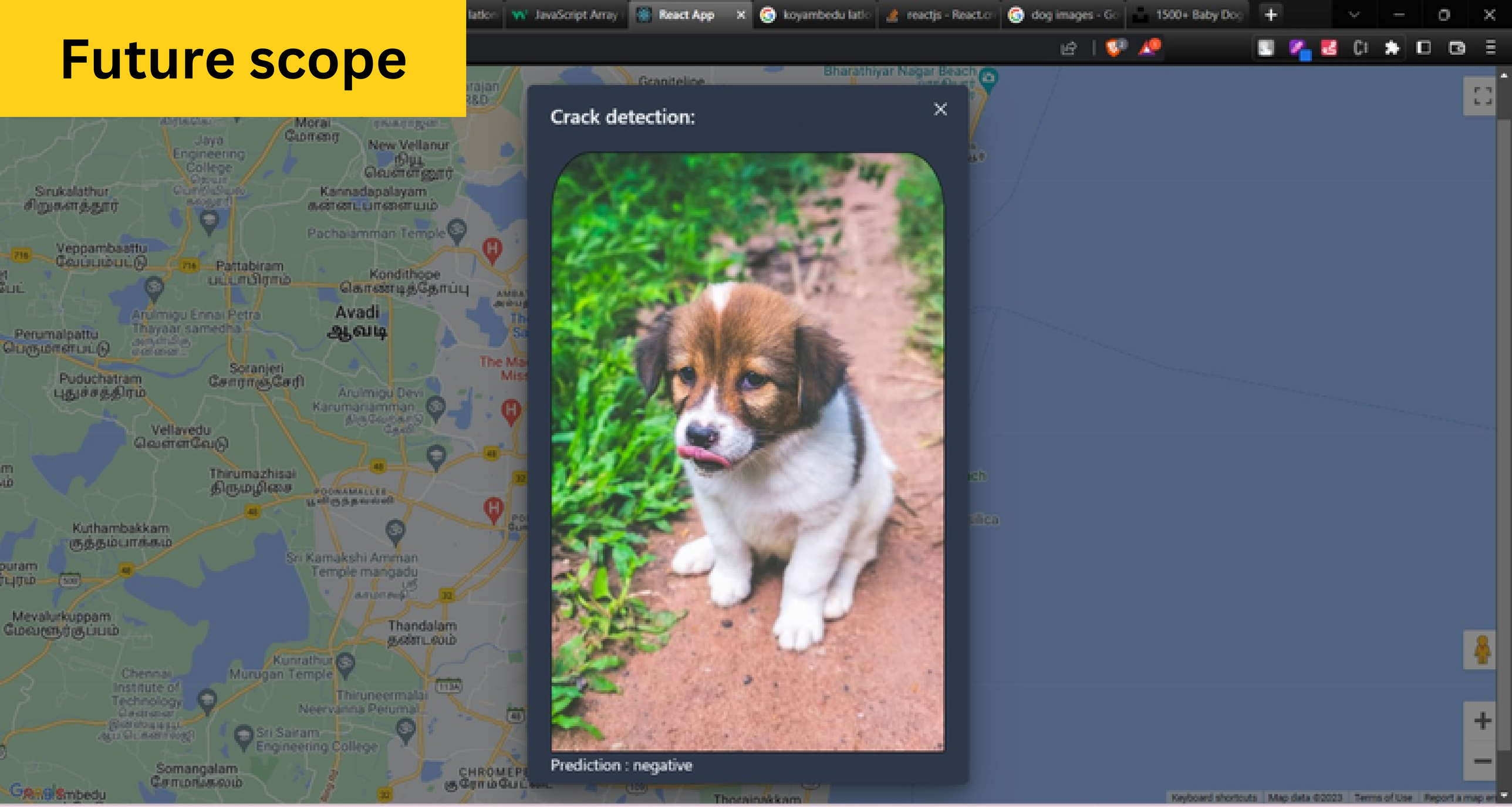
Features

- We designed the app deploying the trained model which includes all the necessary features like prediction, time and location of image capture.

- Using location data, we can analysis the patterns in the crack and avoid bridge failure in the future and using time-location tag feature, we can further look upon factors affecting the development of crack and also analysis large scale such crack pattern in and around the terrain.

- This can be further extended to be stored in centralized server for which website can be created and used in near future for further production enhancement.

Future scope



Innovative

THANK YOU.

We acknowledge our sincere thanks to the hackathon organizers who gave us an opportunity to work hands on, in image processing and classification techniques.

Entrepreneurial
Empowered