An Application for Converting Facial Emotions to Emoticons.

Getting to know each emotion
with Emoticons

แง๊นๆ

MEET THE TEAM

FIN NAK BID

นายกริชลักษณ์	อนันต์สิริวุฒิ	6434402423
นายกัณฑ์พจน์	ลิขิตยิ่งวรา	6434405323
นายเขมภัทร	ภูมา	6434409923
นายจิรพัฒน์	ธนสุทธิวัฒน์	6434412723
นางสาวพัชรมัย	สหสินธพ	6434457023
นางสาวพัณณิต์ตา	ทั่งทอง	6434458623

แง๊นๆ

DATASET

kaggle

Search

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Your Work

VIEWED

Facial Expression R...

MANAS SAMBARE · UPDATED 4 YEARS AGO

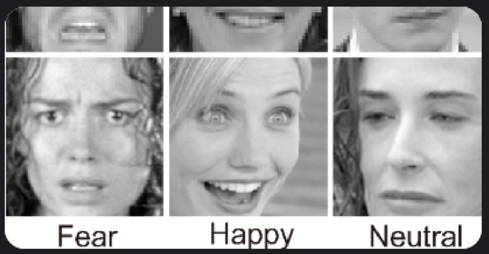
1034

New Notebook

Download (63 MB)

FER-2013

Learn facial expressions from an image



Data Card | Code (332) | Discussion (6) | Suggestions (0)

About Dataset

The data consists of 48×48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image.

The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

Usability
7.50

License
Database: Open Database, Cont...

Expected update frequency
Not specified

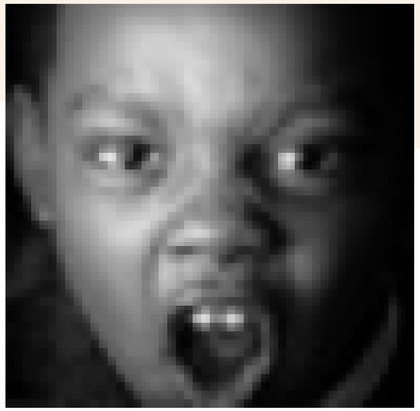
Tags

DATASET

surprise

fear

angry



neutral

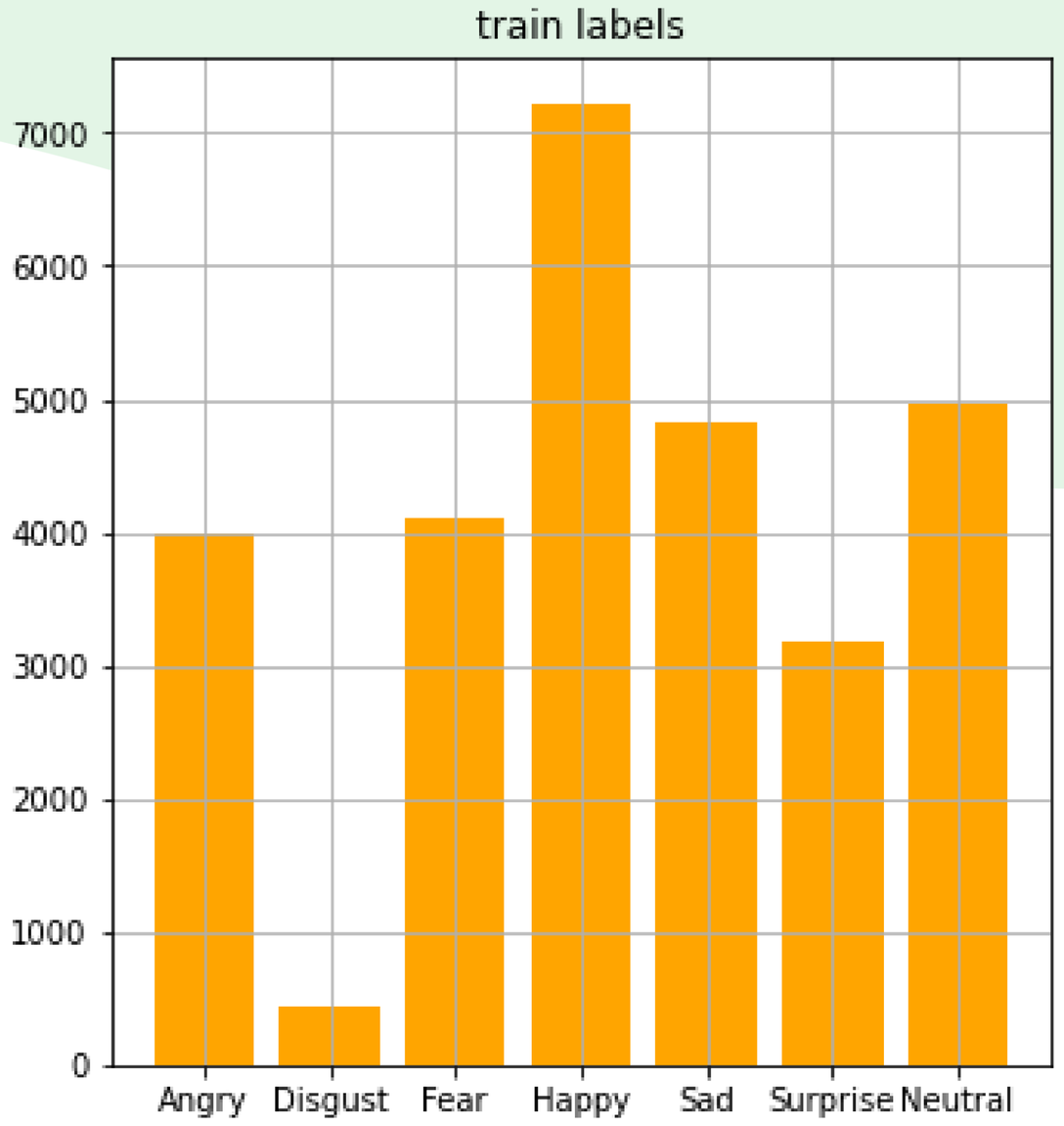
sad

disgust

happy

DATA PREPROCESSING

Disgust -> Angry



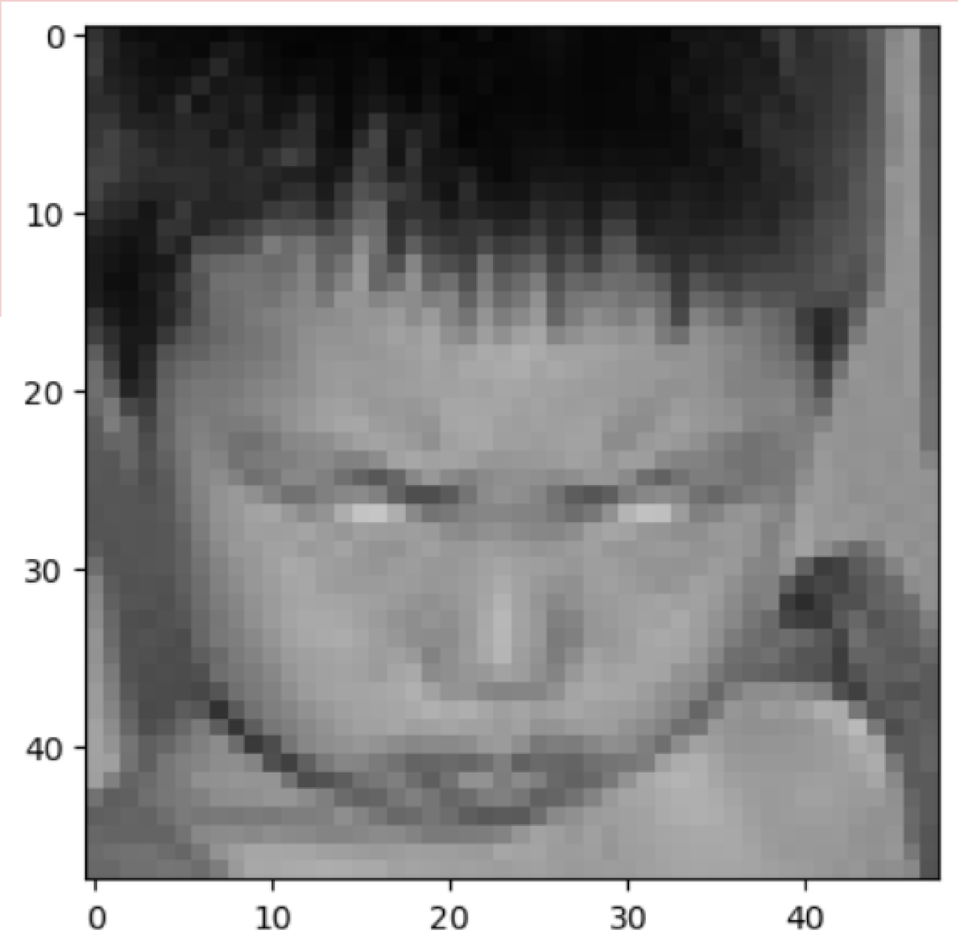
DATA PREPROCESSING

48 * 48 pixels

(48, 48, 1)

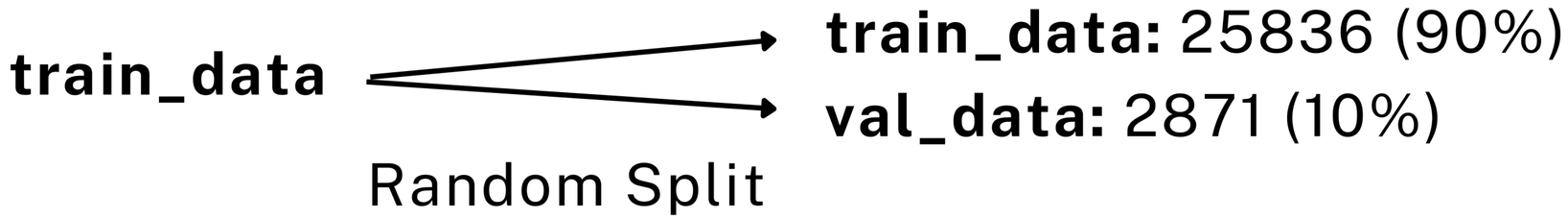
for train set

+ RandomHorizontalFlip()
+ RandomCrop()



DATA PREPROCESSING

```
# Load all the imgs within the specified folder and apply different augmentation
train_data = datasets.ImageFolder(cfg.trainDirectory, transform=train_transform)
test_data = datasets.ImageFolder(cfg.testDirectory, transform=test_transform)
```



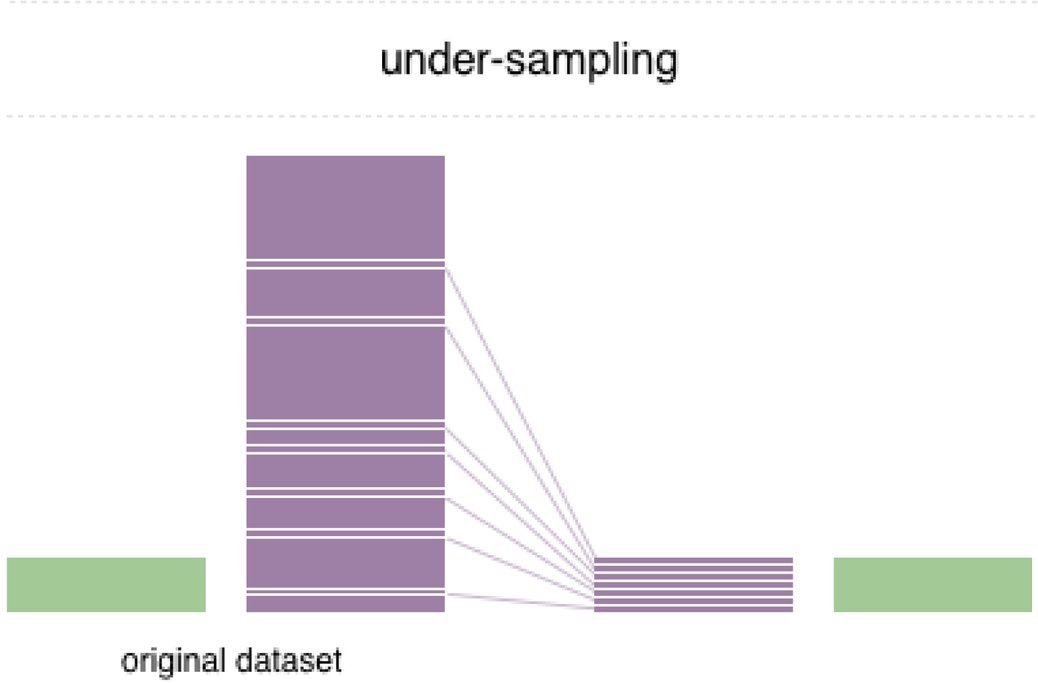
Train Samples

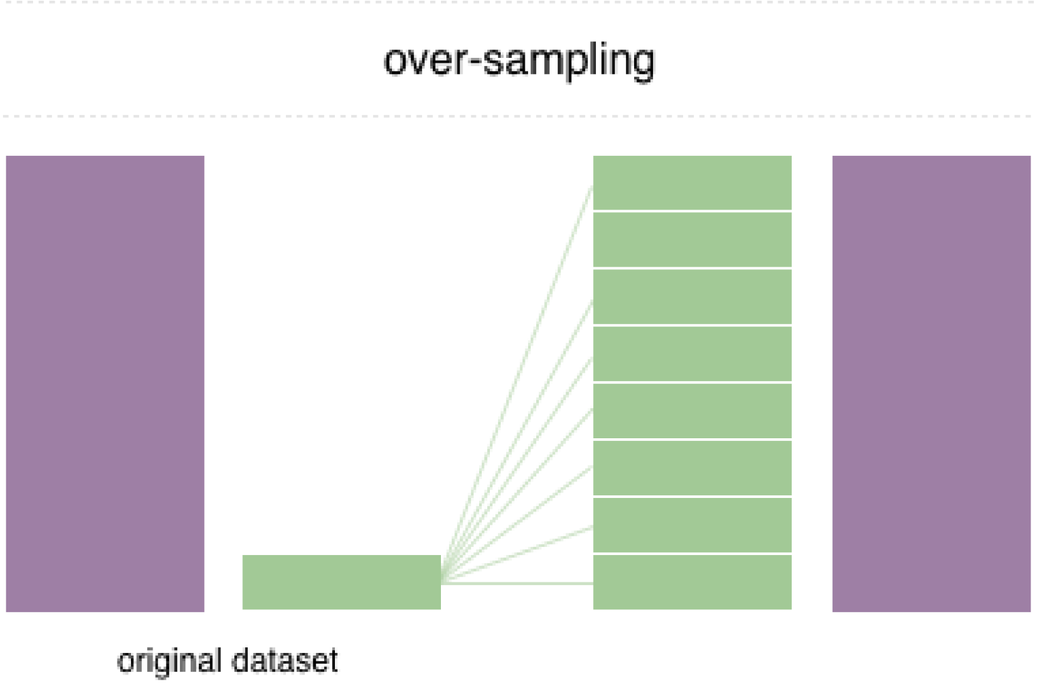
Happy: 6455
Neutral: 4500
Sad: 4333
Angry: 3982
Fear: 3731
Surprise: 2835

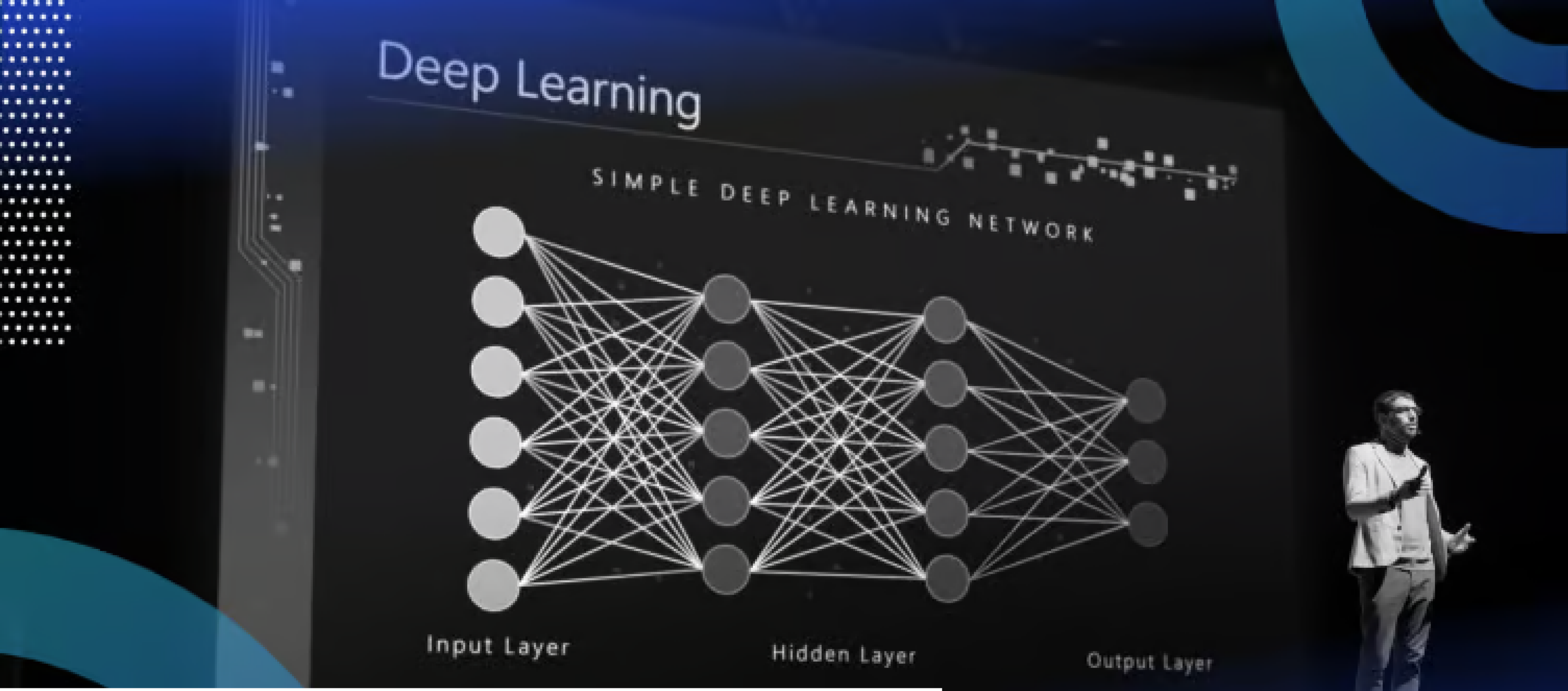
DATA PREPROCESSING

```
class_weight = torch.Tensor([len(train_classes) / c
                             for c in pd.Series(class_count).sort_index().values])
```

```
sampler = WeightedRandomSampler(weights=sample_weight, num_samples=len(train_data), replacement=True)
```







DEEP LEARNING MODEL

DEEP LEARNING MODEL

Created By My Team

emotionNet

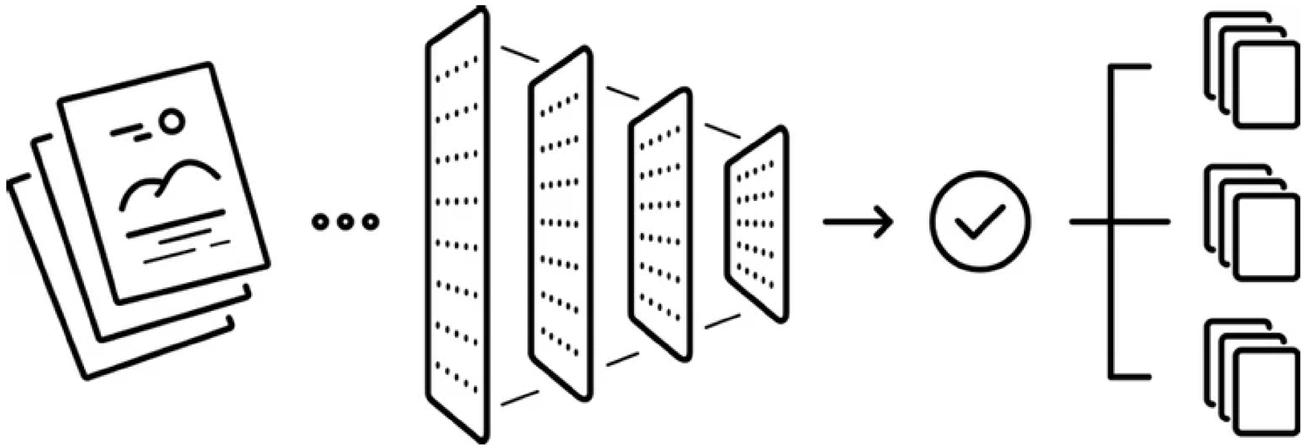
Transfer Learning

mobileNet_V2

resNet18

resNet50

EMOTIONNET



---

feature extraction (features)

Input 48 x 48x 1

--> Conv2d --> BatchNorm2d --> ELU- -> Conv2d --> BatchNorm2d --> ELU --> MaxPool2d-->
48 x 48x 32 " " " " " 24 x 24x 32

--> Conv2d --> BatchNorm2d --> ELU- -> Conv2d --> BatchNorm2d --> ELU --> MaxPool2d-->
24 x 24x 64 " " " " " 12 x 12x 64

--> Conv2d --> BatchNorm2d --> ELU- -> Conv2d --> BatchNorm2d --> ELU --> MaxPool2d-->
12 x 12x 128 " " " " " 6 x 6 x 128

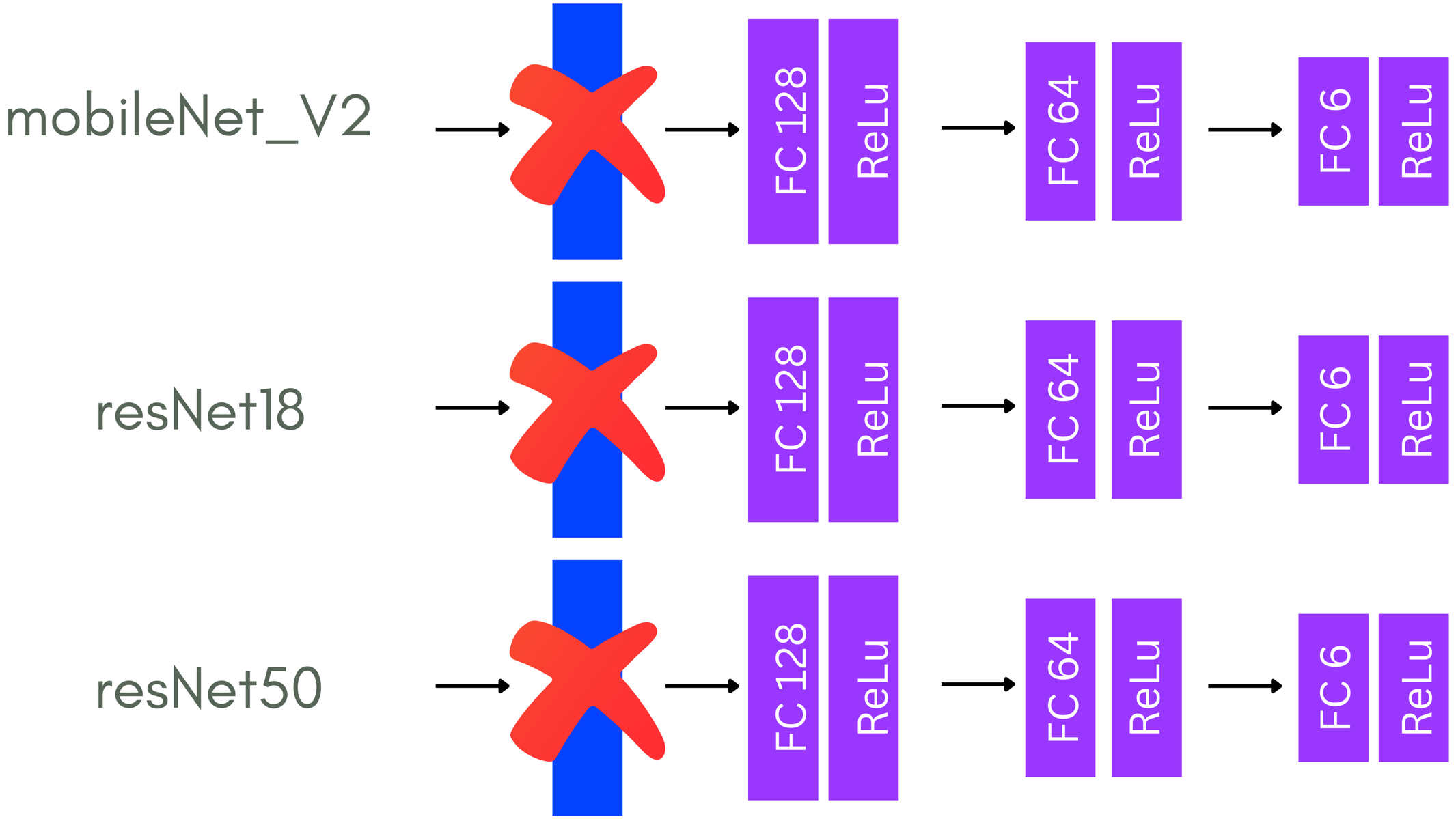
-> Flatten (6 x 6 x 128)

---

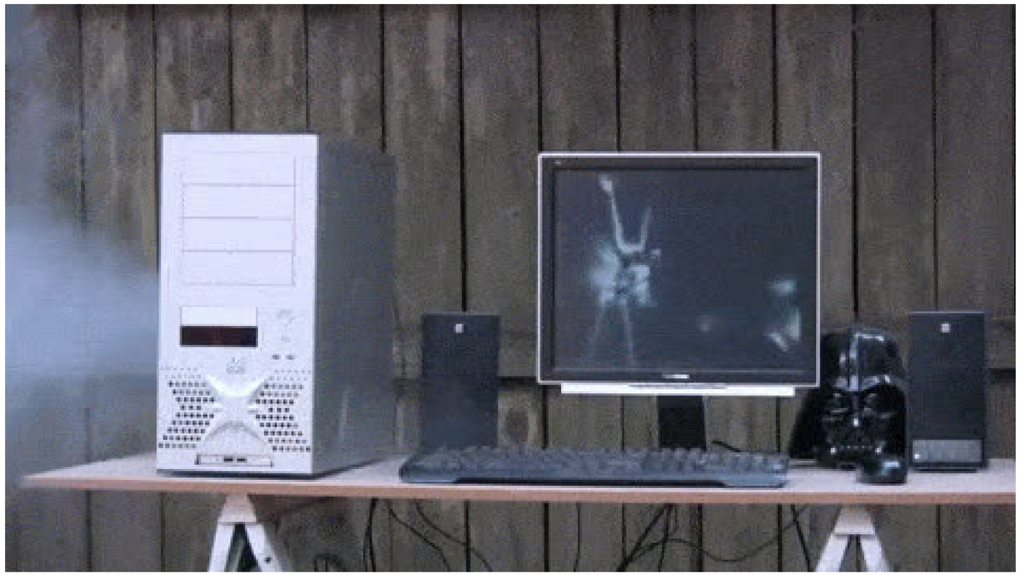
classification (classifier)

--> Linear --> ELU --> Dropout --> Linear (num_of_classes) -->
64 64 64(random drop) 6

TRANSFER LEARNING



TRAINING



GPU

Epochs: 100

Batch_size: 16

Optimizer: SGD

Learning Rate: 0.1

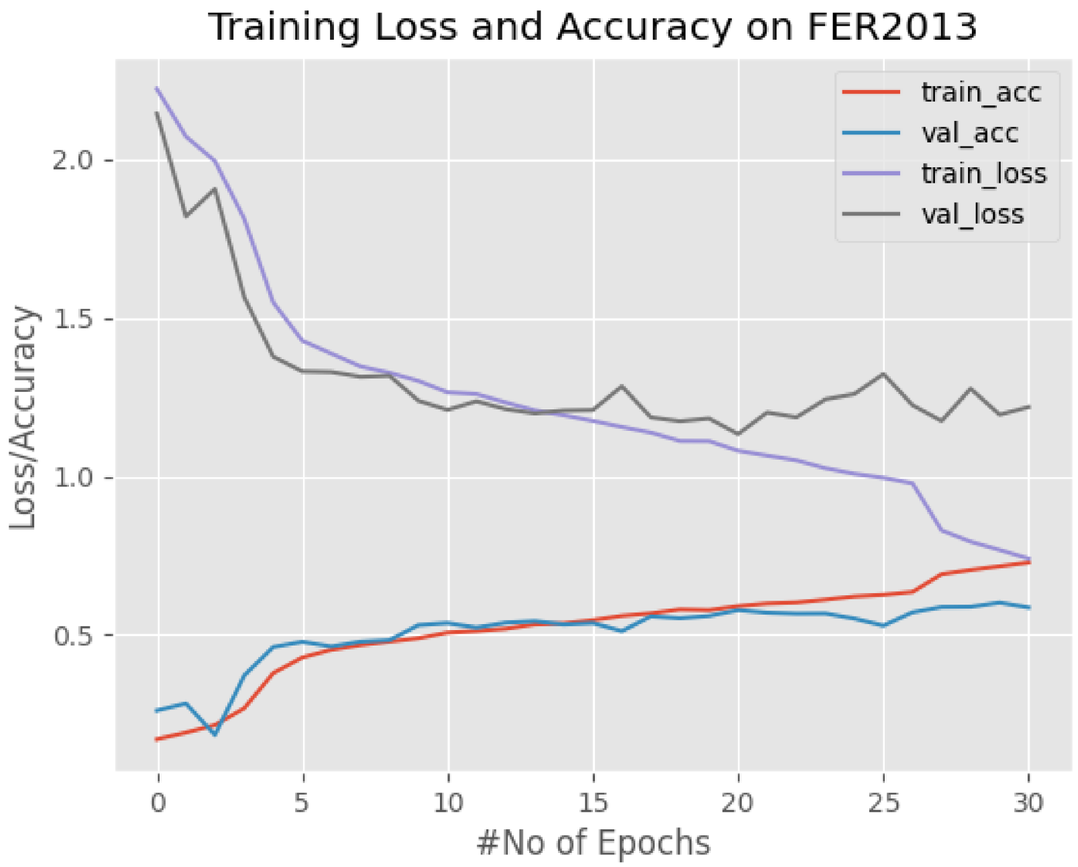
Loss Function: CrossEntropyLoss

LRScheduler

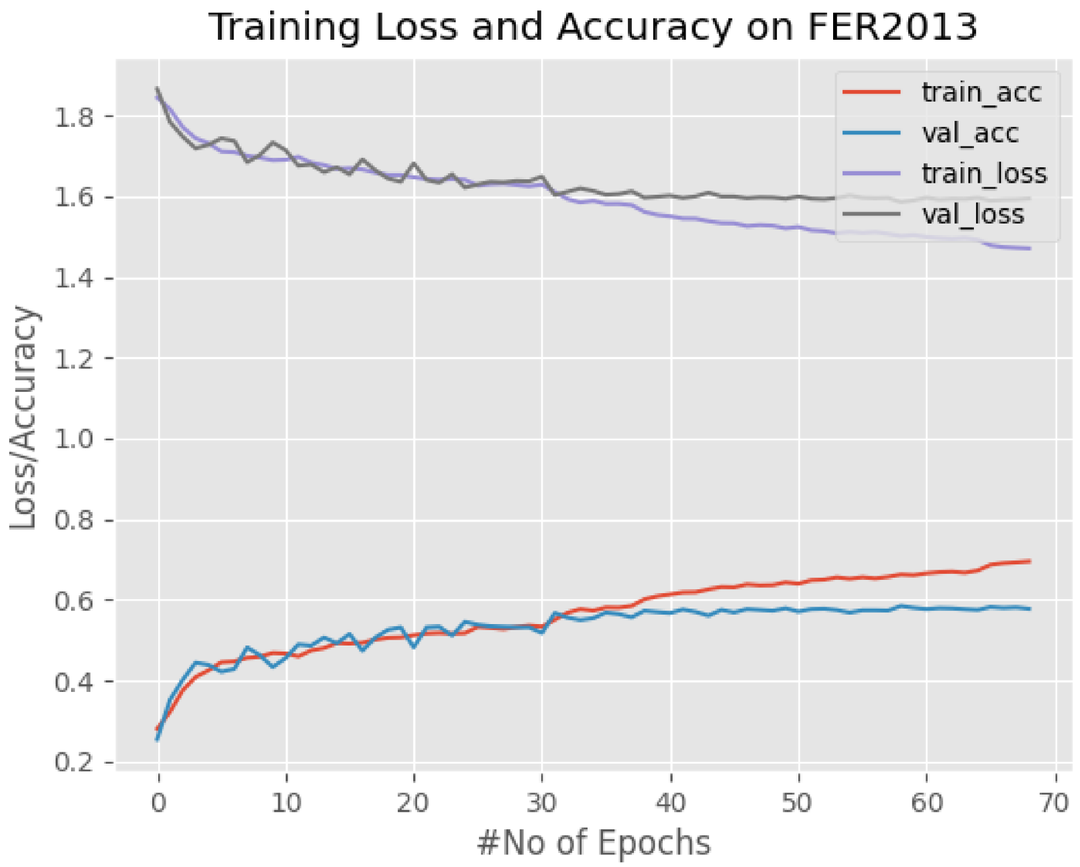
Early_Stopping

EVALUATION

EmotionNet

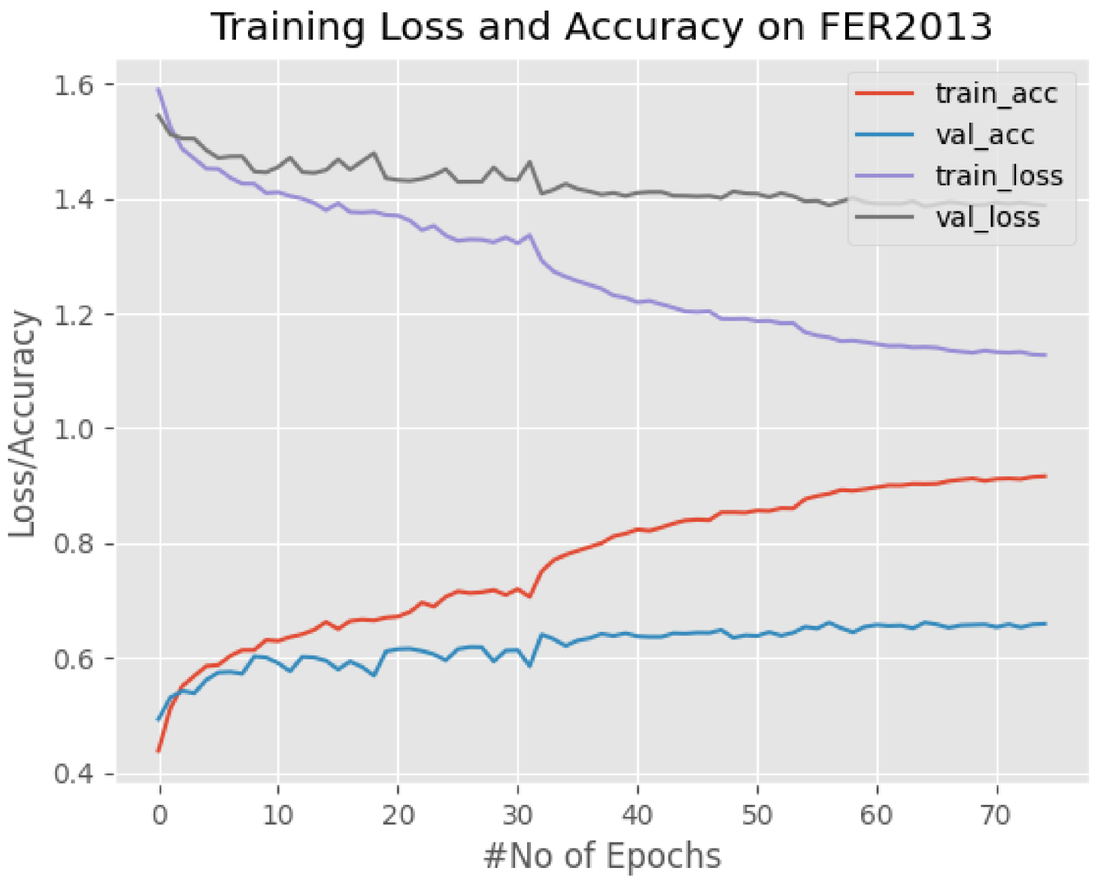


MobileNet_v2

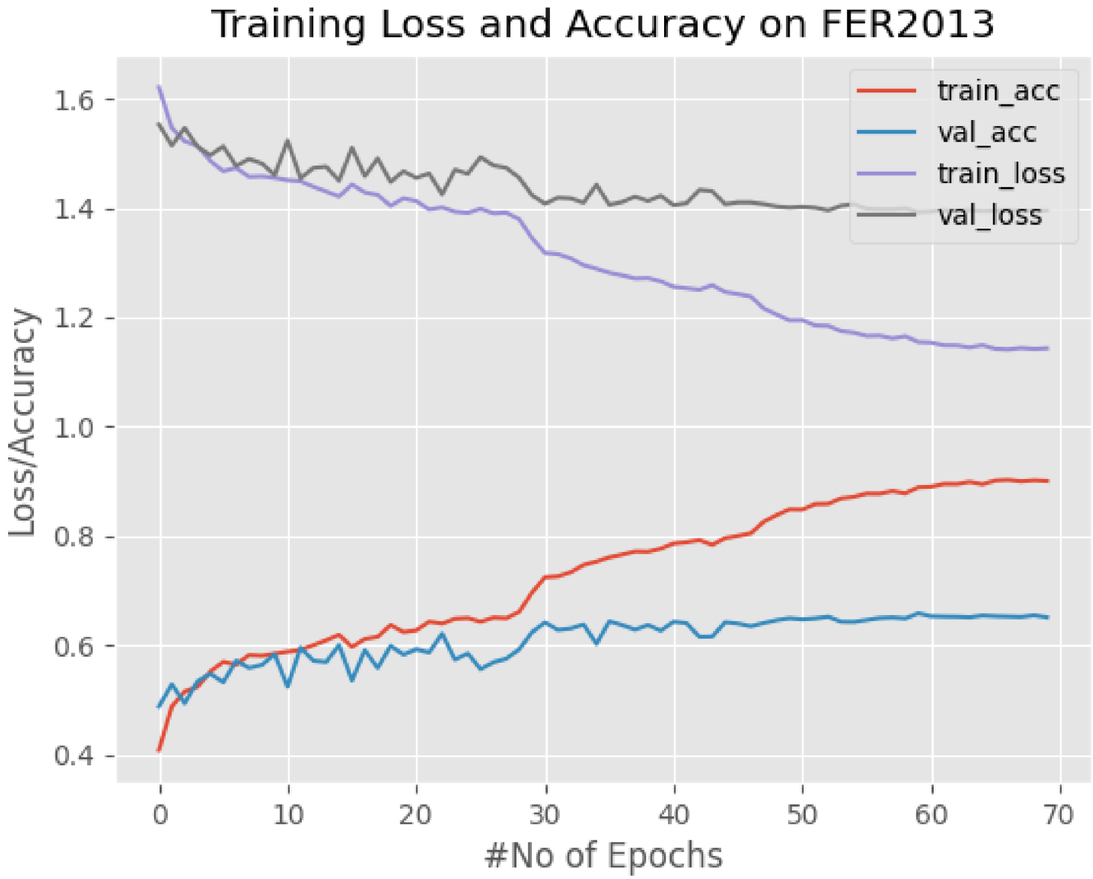


EVALUATION

ResNet18



ResNet50



EVALUATION

EmotionNet

	precision	recall	f1-score	support
angry	0.52	0.52	0.52	1069
fear	0.37	0.47	0.41	1024
happy	0.85	0.73	0.79	1774
neutral	0.53	0.48	0.50	1233
sad	0.43	0.51	0.46	1247
surprise	0.79	0.65	0.71	831
accuracy			0.57	7178
macro avg	0.58	0.56	0.57	7178
weighted avg	0.60	0.57	0.58	7178

EVALUATION

MobileNet_V2

	precision	recall	f1-score	support
angry	0.55	0.58	0.56	1069
fear	0.37	0.21	0.27	1024
happy	0.83	0.78	0.80	1774
neutral	0.51	0.61	0.56	1233
sad	0.44	0.51	0.48	1247
surprise	0.72	0.75	0.73	831
accuracy			0.59	7178
macro avg	0.57	0.58	0.57	7178
weighted avg	0.59	0.59	0.58	7178

EVALUATION

ResNet18

	precision	recall	f1-score	support
angry	0.61	0.64	0.63	1069
fear	0.52	0.48	0.50	1024
happy	0.83	0.86	0.85	1774
neutral	0.58	0.66	0.62	1233
sad	0.55	0.49	0.52	1247
surprise	0.81	0.76	0.78	831
accuracy			0.66	7178
macro avg	0.65	0.65	0.65	7178
weighted avg	0.66	0.66	0.66	7178

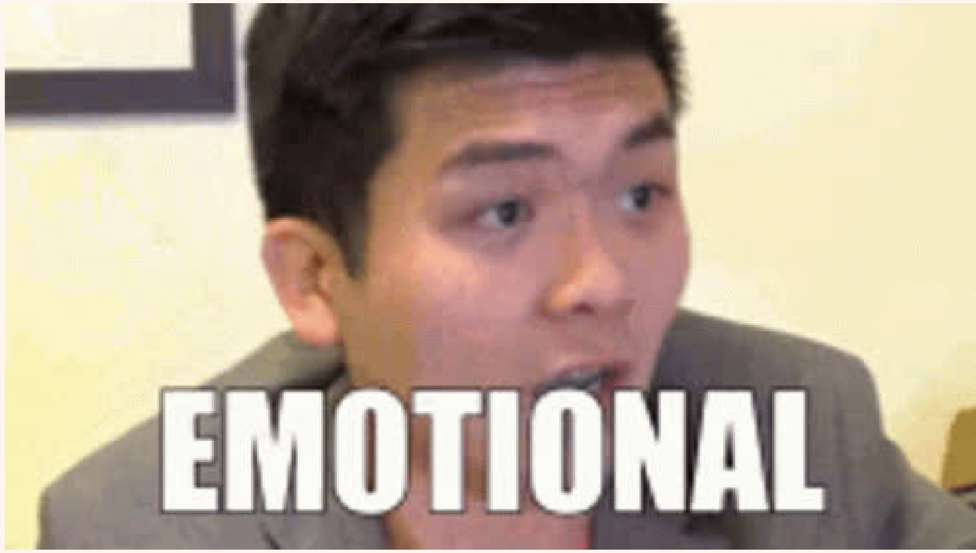
EVALUATION

ResNet50

	precision	recall	f1-score	support
angry	0.62	0.59	0.61	1069
fear	0.51	0.44	0.47	1024
happy	0.84	0.85	0.85	1774
neutral	0.56	0.63	0.59	1233
sad	0.52	0.51	0.52	1247
surprise	0.75	0.79	0.77	831
accuracy			0.65	7178
macro avg	0.63	0.63	0.63	7178
weighted avg	0.65	0.65	0.65	7178

REAL WORLD USE

emotionNet

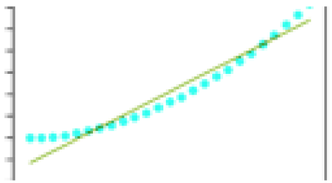


mobileNet_V2

resNet18

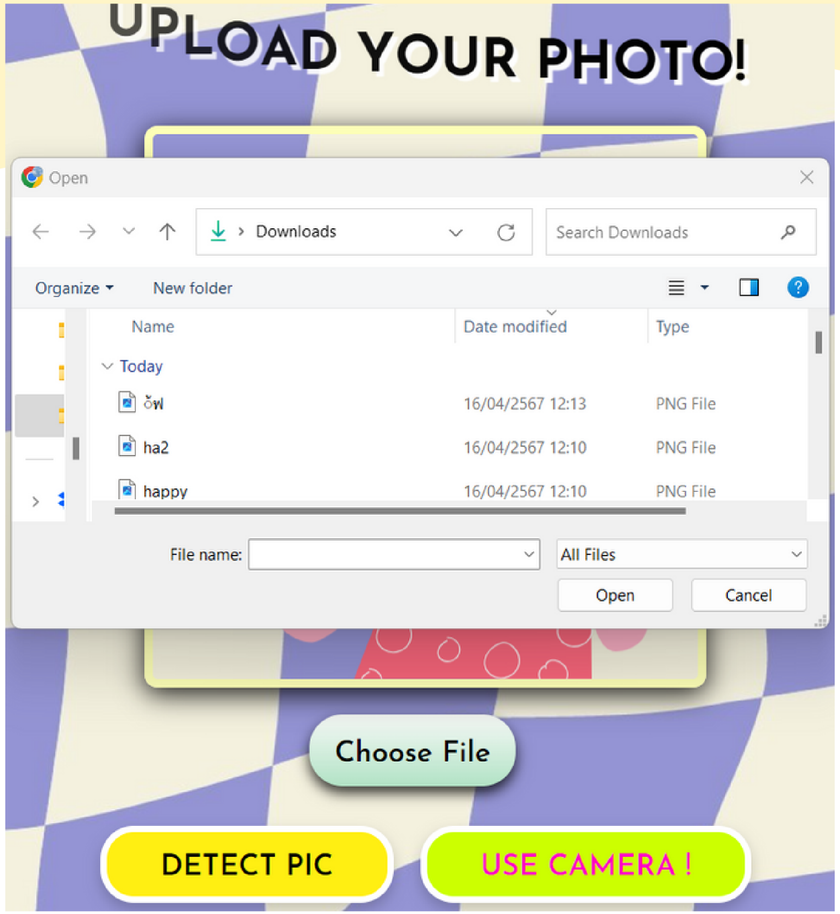
resNet50

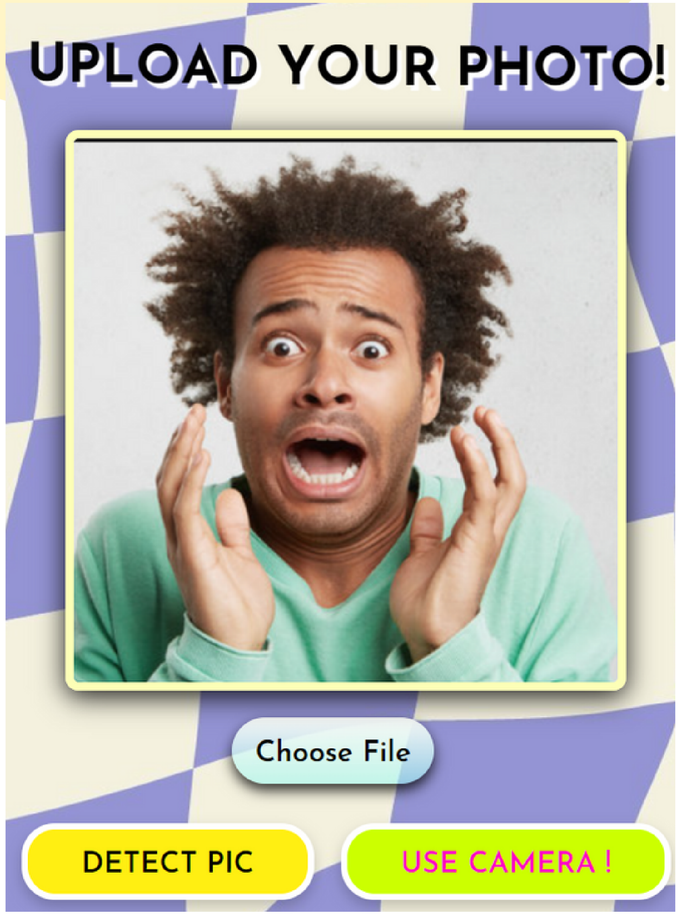
DEPLOYING MODEL



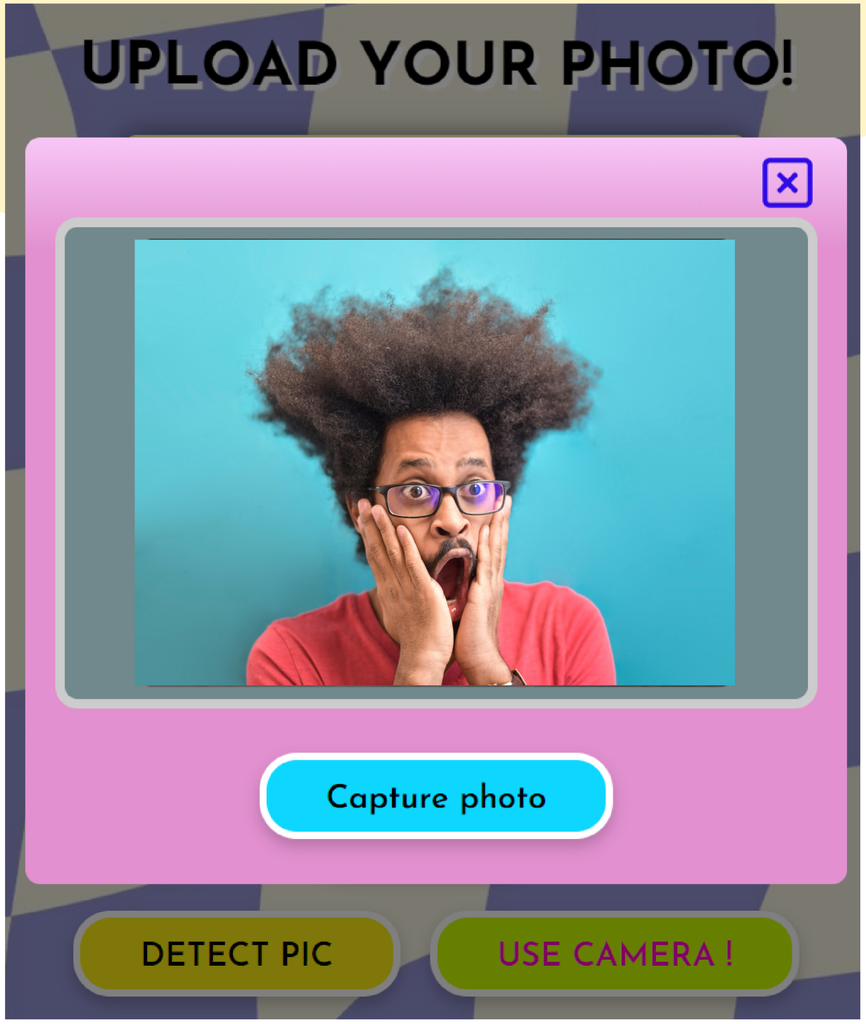
Front end

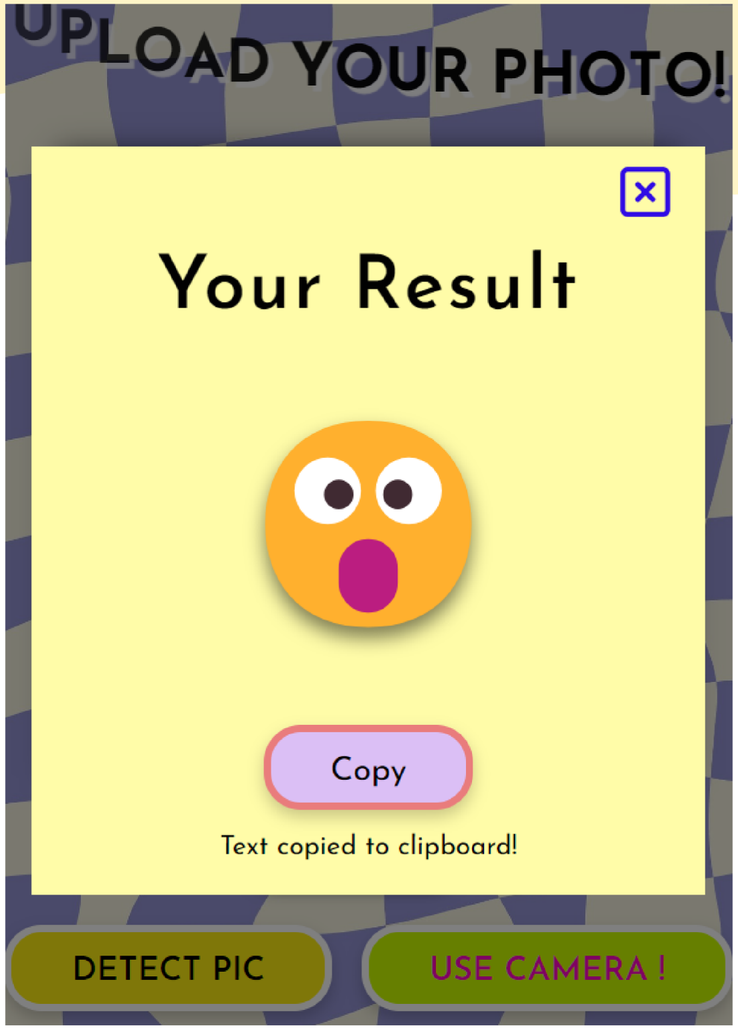

UPLOAD YOUR PHOTO!
Choose File
DETECT PIC
USE CAMERA !


UPLOAD YOUR PHOTO!
Open
Downloads
Search Downloads
Organize
New folder
Name
Date modified
Type
Today
16/04/2567 12:13
PNG File
ha2
16/04/2567 12:10
PNG File
happy
16/04/2567 12:10
PNG File
File name:
All Files
Open
Cancel
Choose File
DETECT PIC
USE CAMERA !

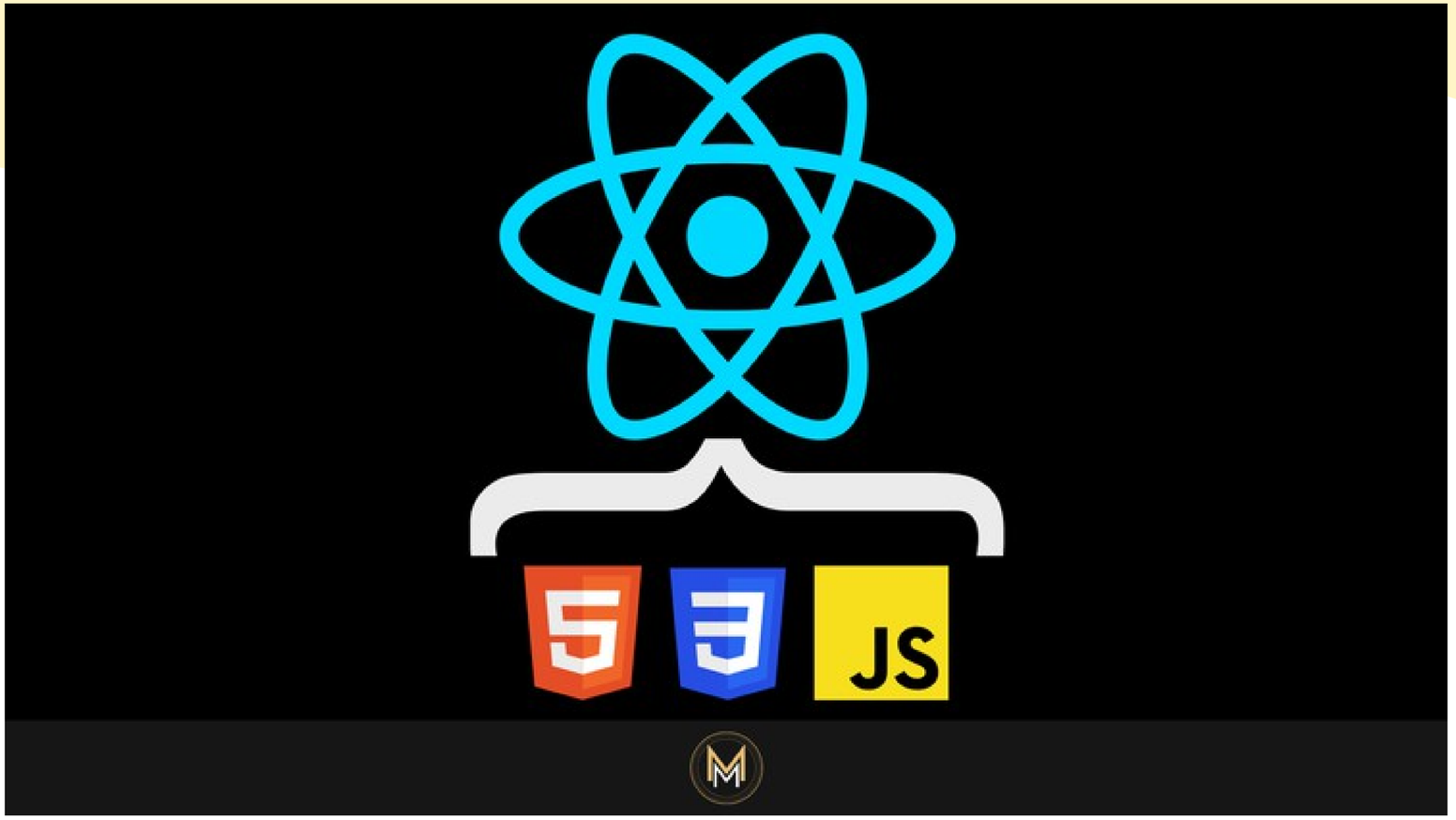

UPLOAD YOUR PHOTO!
Choose File
DETECT PIC
USE CAMERA !

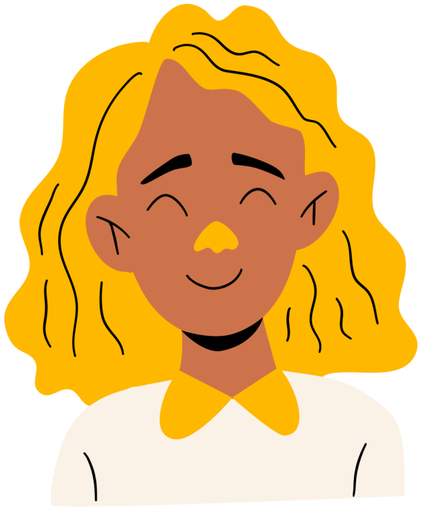
Front end


UPLOAD YOUR PHOTO!
Capture photo
DETECT PIC
USE CAMERA !


UPLOAD YOUR PHOTO!
Your Result
Copy
Text copied to clipboard!
DETECT PIC
USE CAMERA !

React Framework





React Libraries

Axios

react-axios

Axios Component for React with child function callback. This is intended to allow in render async requests.

Features

- Same great features found in Axios
- Component driven
- Child function callback ***(error, response, isLoading, makeRequest, axios) => { }***
- Auto cancel previous requests
- Debounce to prevent rapid calls.
- Request only invoked on prop change and *isReady* state.
- Callback props for ***onSuccess***, ***onError***, and ***onLoading***
- Supports custom axios instances through ***props*** or a ***<AxiosProvider ... >***
- Create your own request components wrapped using the ***withAxios({options})(ComponentToBeWrapped)*** HoC

Webcam

react-webcam

build passing downloads 197k/week

React Webcam

DEMO: https://codepen.io/mozmorris/pen/JLZdoP

https://www.npmjs.com/package/react-webcam

Webcam component for React. See http://caniuse.com/#feat=stream for browser compatibility.

Note: Browsers will throw an error if the page is loaded from insecure origin. I.e. Use https.

THANK YOU
THANK YOU

THANK YOU
THANK YOU