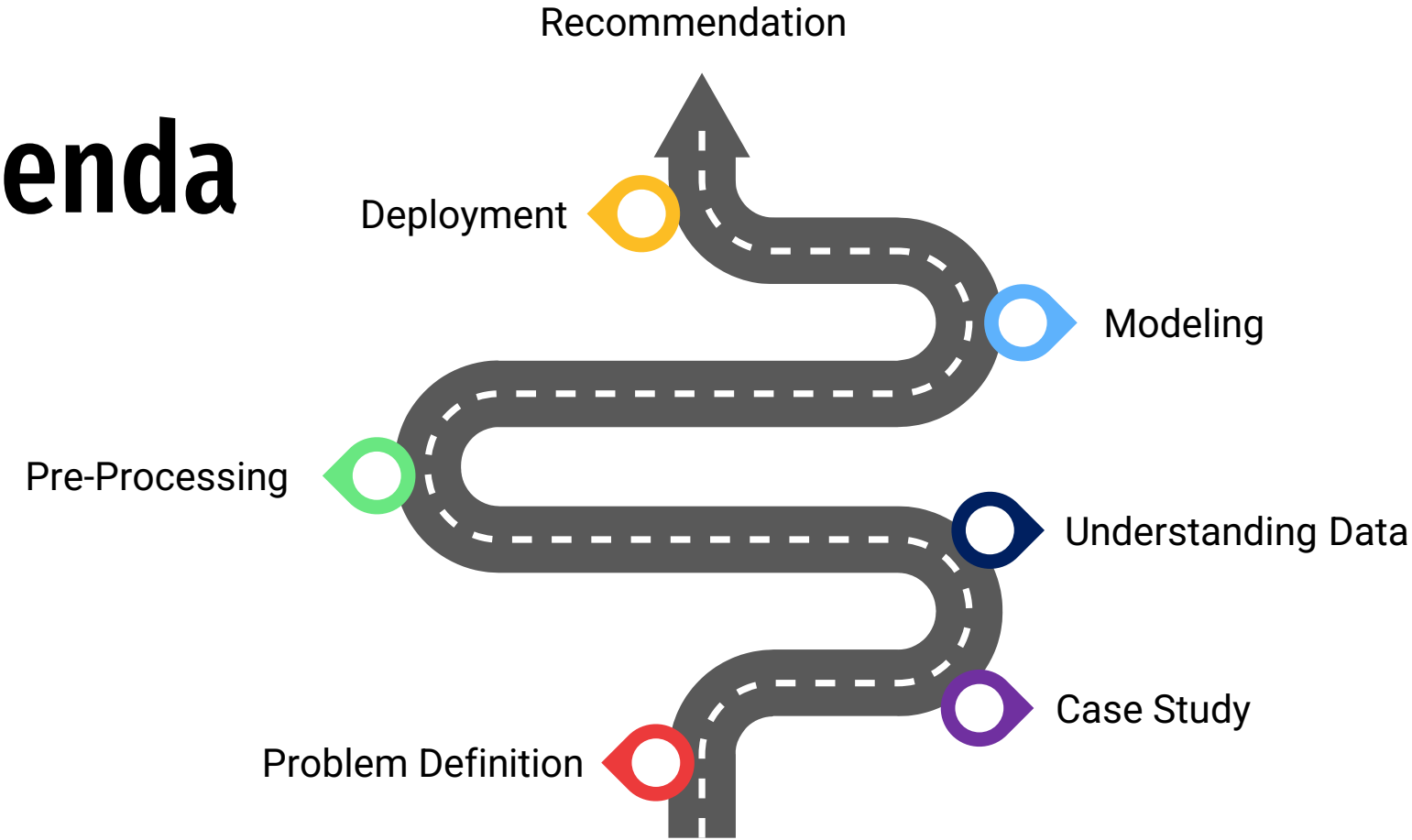
SAMSUNG

Samsung Innovation Campus

Artificial Intelligence Course

Together for Tomorrow!
Enabling People
Education for Future Generations

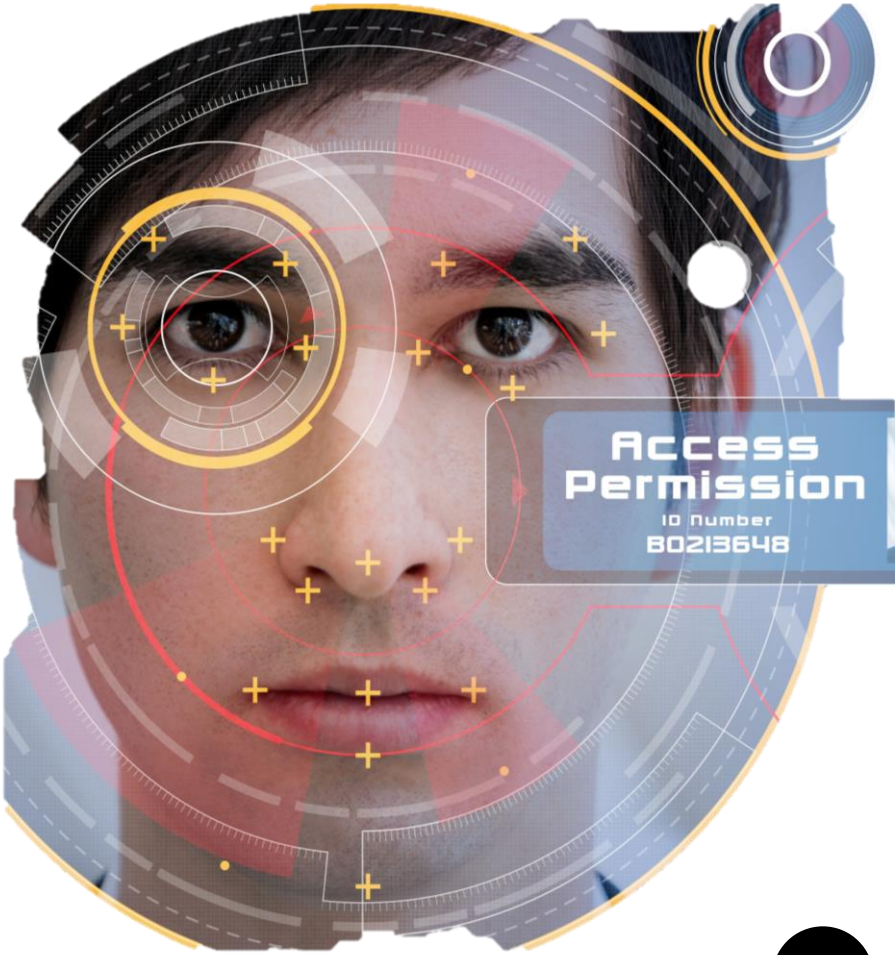
Agenda


Recommendation
Deployment
Modeling
Pre-Processing
Understanding Data
Case Study
Problem Definition

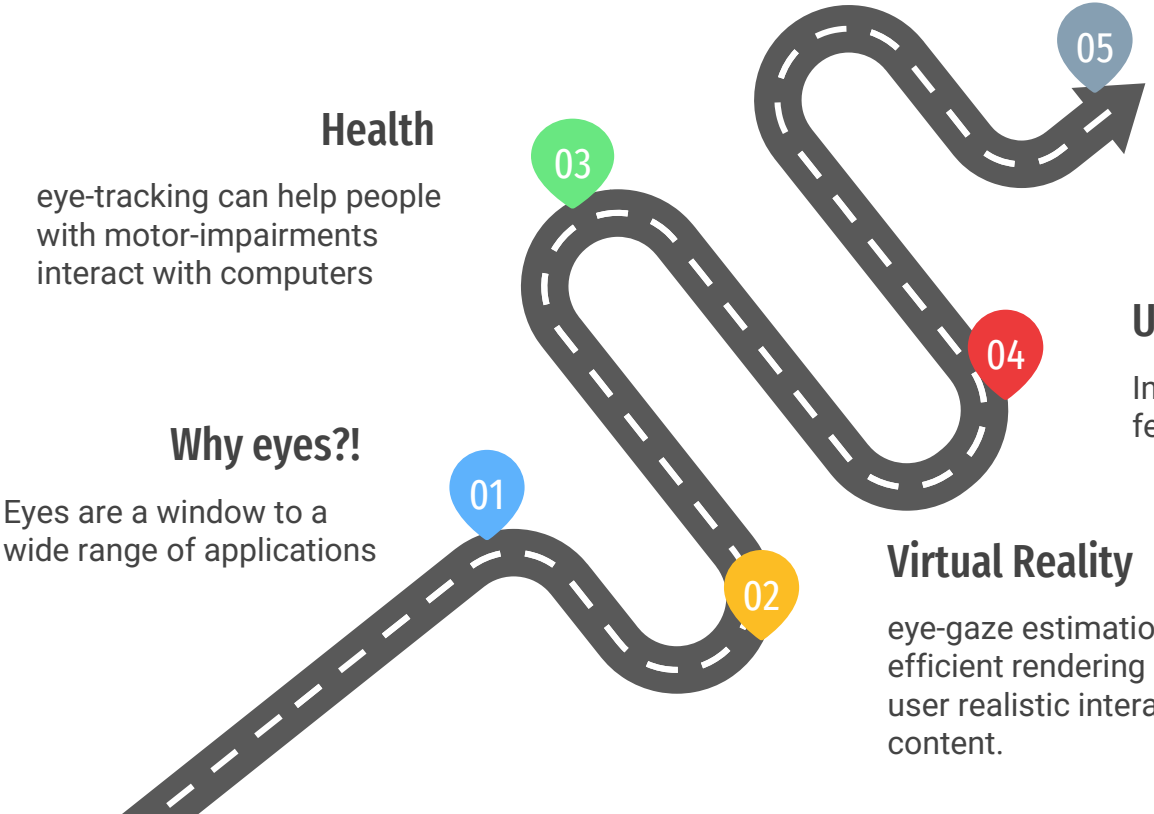
The Detection Of Eyes

“The eyes are the mirror of the soul and reflect everything that seems to be hidden.”

Paulo Coelho



Problem Definition



Why eyes?!

Eyes are a window to a wide range of applications

Health

eye-tracking can help people with motor-impairments interact with computers

Virtual Reality

eye-gaze estimation to enable efficient rendering and to offer the user realistic interaction with the content.

UI & UX

Insight into which features on a web page.

Automotive field

to enhance safety and driver fatigue detection

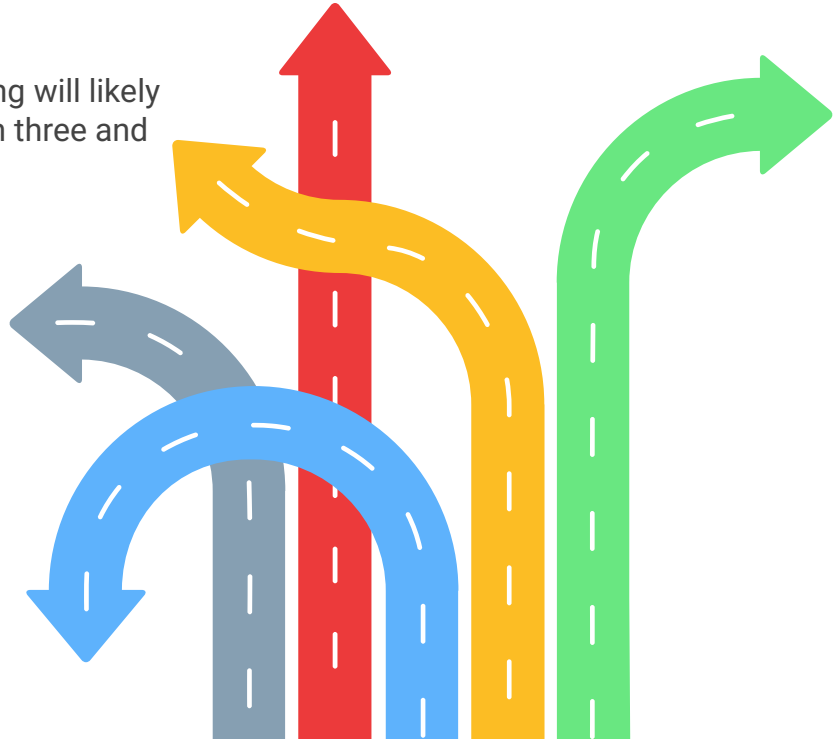
Problem Definition

USA, On average, one pedestrian dies from a car crash every 88 minutes.

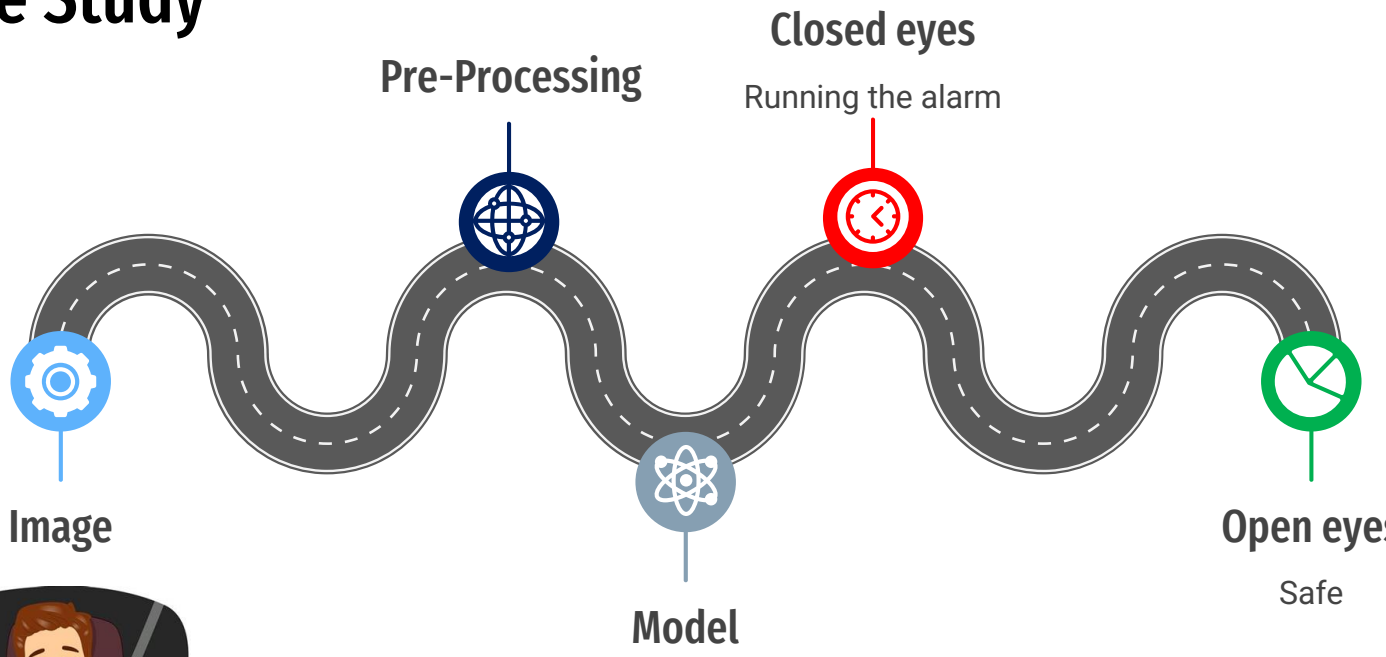
An average human being will likely crash their car between three and four times in a lifetime

Drowsy driving is a significant cause of fatal road crashes. The NHTSA estimates that this factor was responsible for 795 deaths in 2017.

1.35 million people die in car accidents per annum globally



Case Study


Pre-Processing
Closed eyes
Running the alarm
Image
Model
Open eyes
Safe

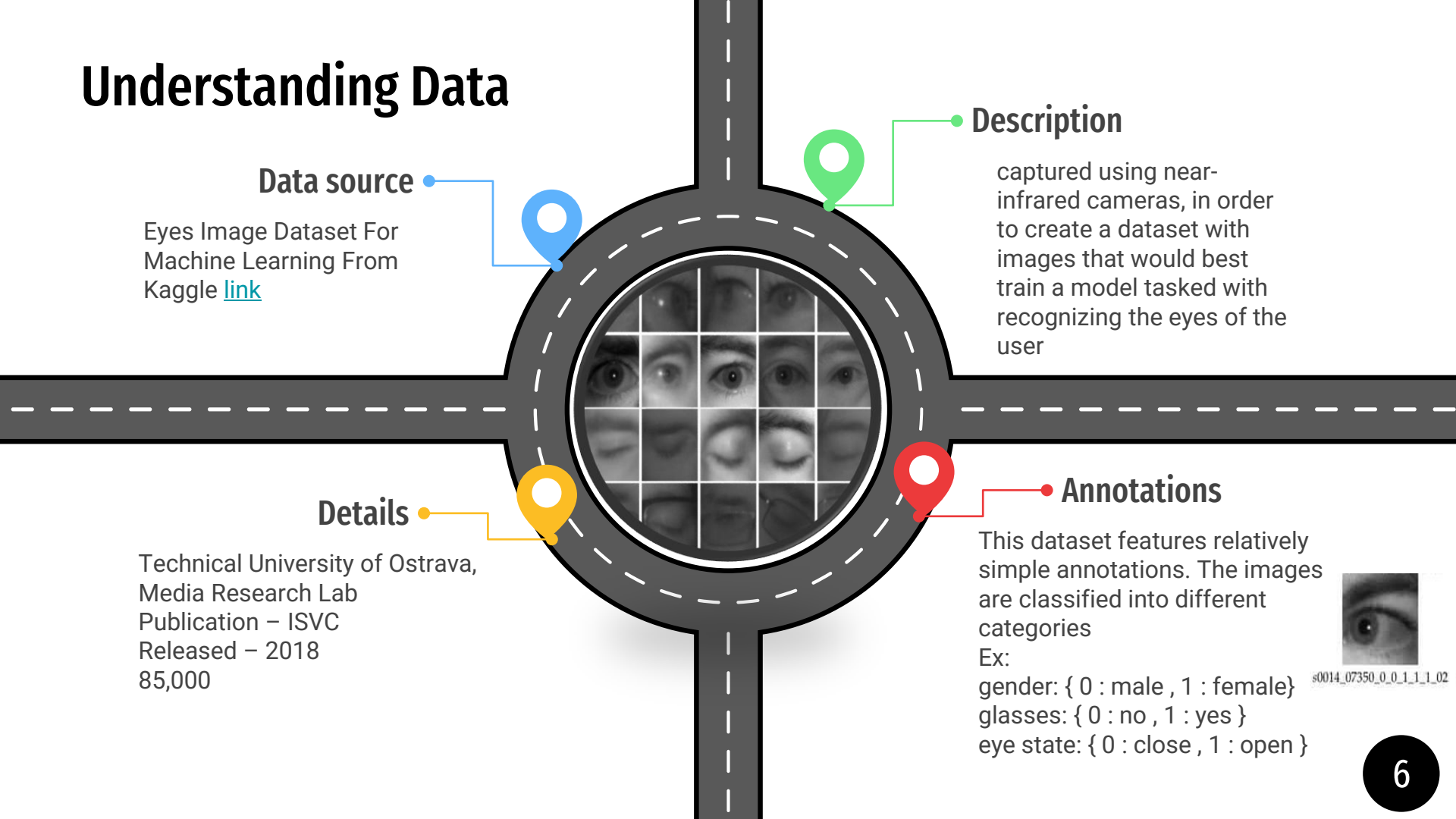
Understanding Data

Data source

Eyes Image Dataset For Machine Learning From Kaggle link

Description

captured using near-infrared cameras, in order to create a dataset with images that would best train a model tasked with recognizing the eyes of the user



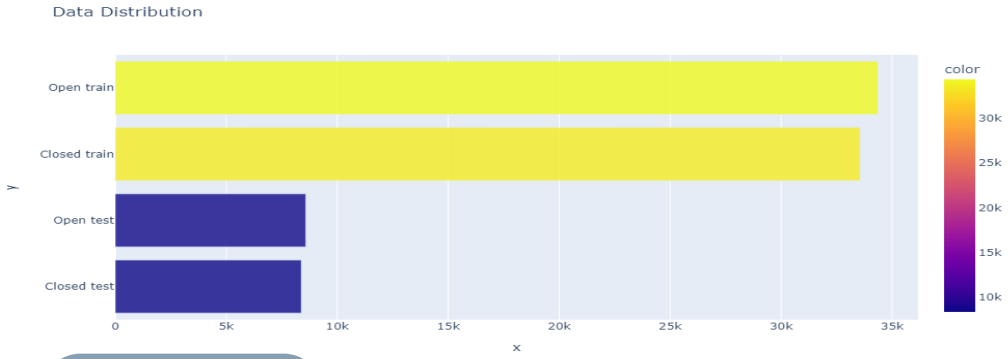
Details

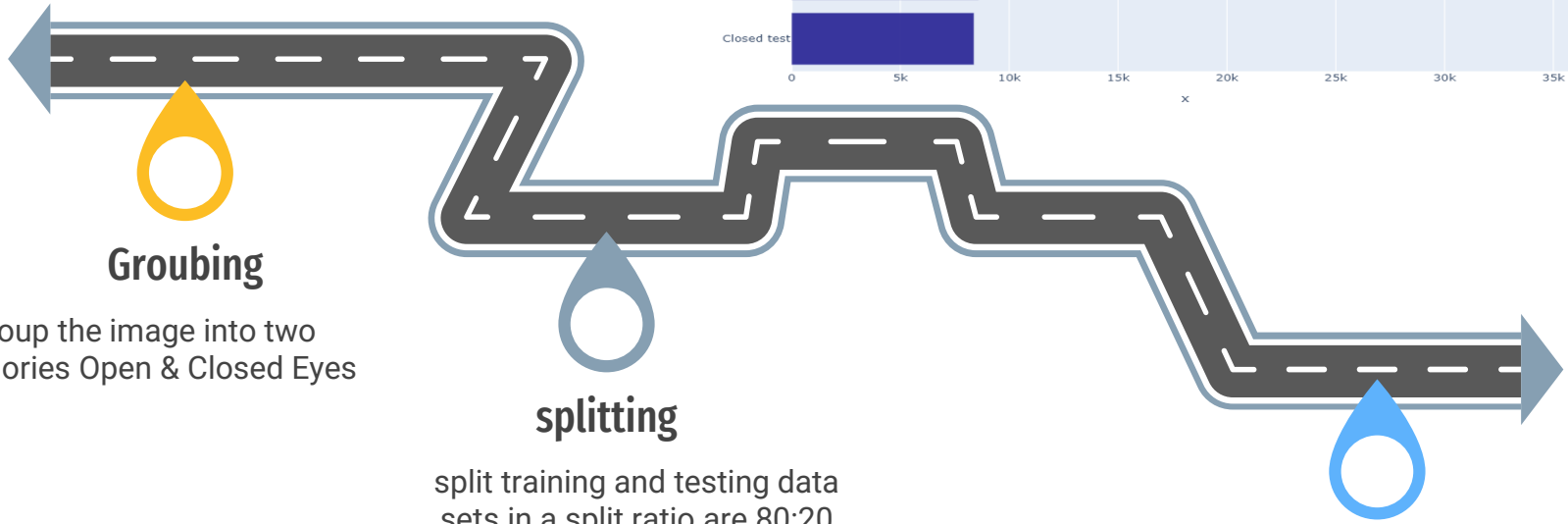
Technical University of Ostrava, Media Research Lab
Publication – ISVC
Released – 2018
85,000

Annotations

This dataset features relatively simple annotations. The images are classified into different categories
Ex:
gender: { 0 : male , 1 : female}
glasses: { 0 : no , 1 : yes }
eye state: { 0 : close , 1 : open }

Pre-Processing





Groubing

Group the image into two categories Open & Closed Eyes

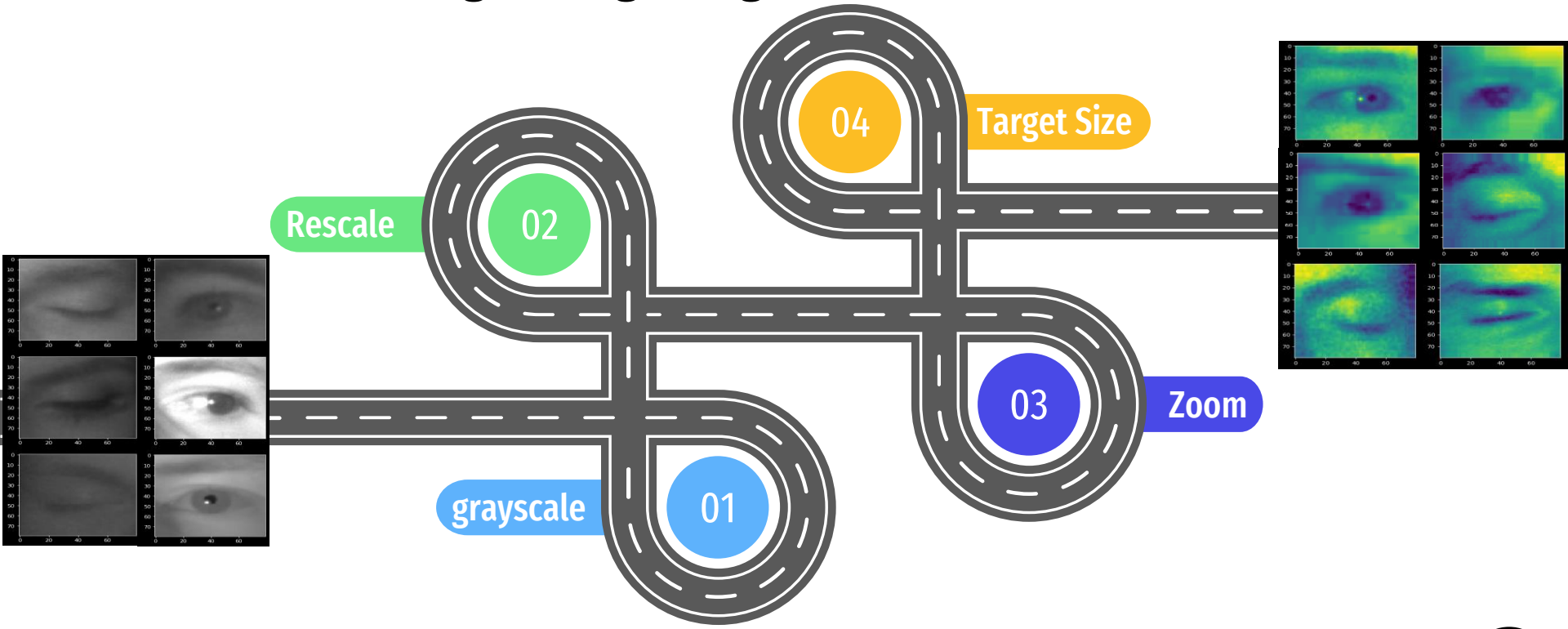
splitting

split training and testing data sets in a split ratio are 80:20

Balanced

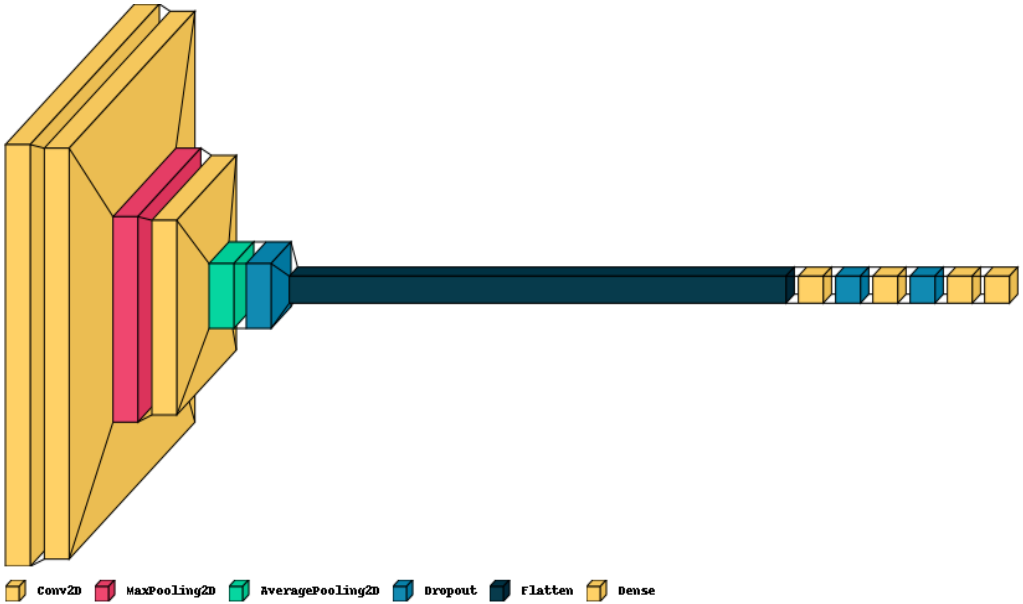
We can say that the number of Open Images and Closed Images in approximately the same.

Pre-Processing Using ImageDataGenerator



Model - CNN

Model Architecture



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 78, 78, 32)	320
conv2d_1 (Conv2D)	(None, 76, 76, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 38, 38, 64)	0
conv2d_2 (Conv2D)	(None, 36, 36, 64)	36928
average_pooling2d (AveragePo	(None, 12, 12, 64)	0
dropout (Dropout)	(None, 12, 12, 64)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 90)	829530
dropout_1 (Dropout)	(None, 90)	0
dense_1 (Dense)	(None, 64)	5824
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 1)	33

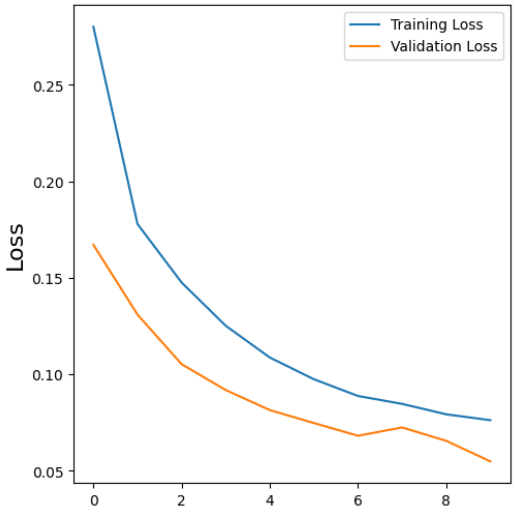
Total params: 893,211
Trainable params: 893,211
Non-trainable params: 0

Model - CNN

Model training

```
Epoch 9/10
1061/1061 [==============] - 100s 94ms/step - loss: 0.0793 - accuracy: 0.9724 - val_loss: 0.0655 - val_accuracy: 0.9757
Epoch 00009: val_loss improved from 0.06815 to 0.06549, saving model to ./data set/data/CNNmodel2.h5

Epoch 10/10
1061/1061 [==============] - 99s 94ms/step - loss: 0.0762 - accuracy: 0.9735 - val_loss: 0.0549 - val_accuracy: 0.9809
Epoch 00010: val_loss improved from 0.06549 to 0.05486, saving model to ./data set/data/CNNmodel2.h5
```



7.6

Loss

97.4

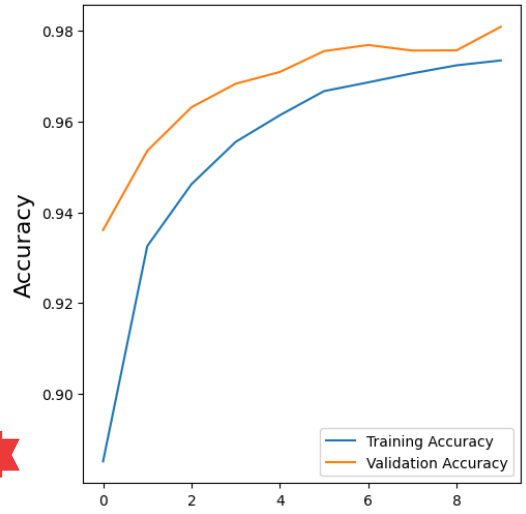
Accuracy

5.4

Val-loss

98

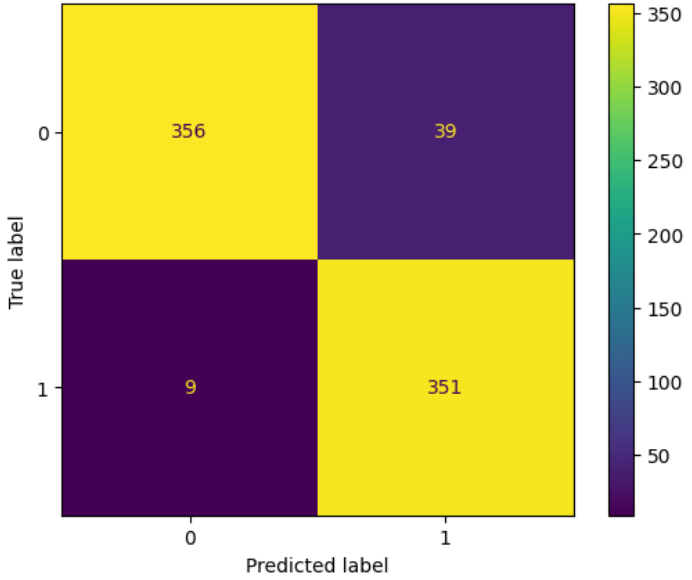
Val-accuracy



Model - CNN

Model Evaluation

	precision	recall	f1-score	support
Closed	0.98	0.90	0.94	395
Open	0.90	0.97	0.94	360
accuracy			0.94	755
macro avg	0.94	0.94	0.94	755
weighted avg	0.94	0.94	0.94	755

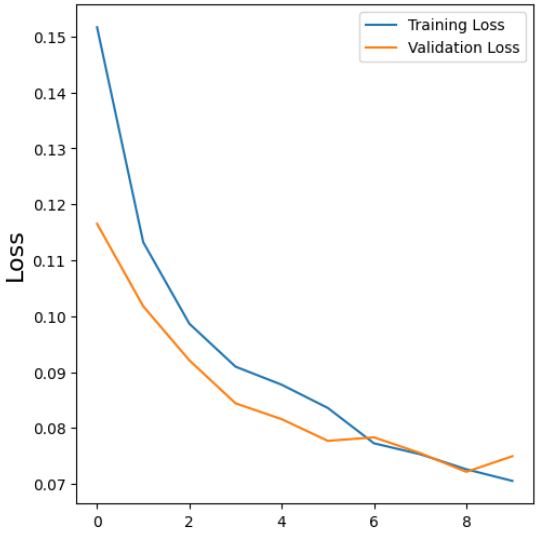


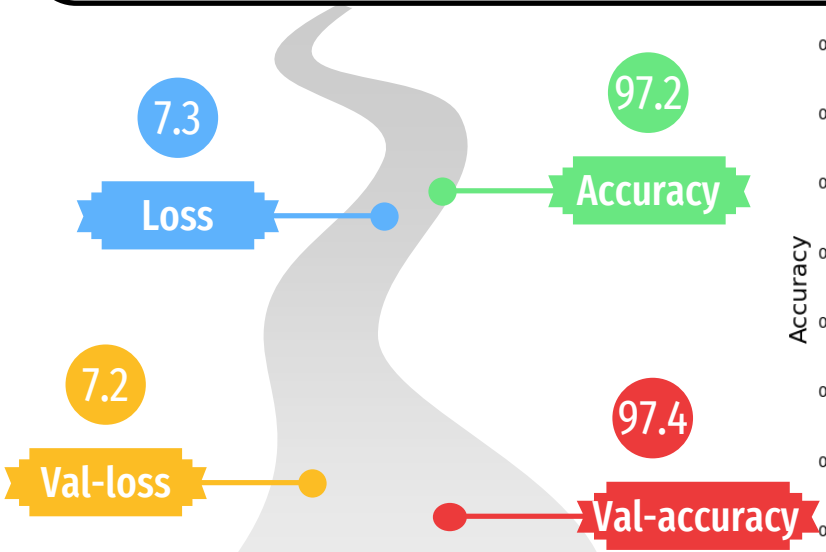
Model - InceptionV3

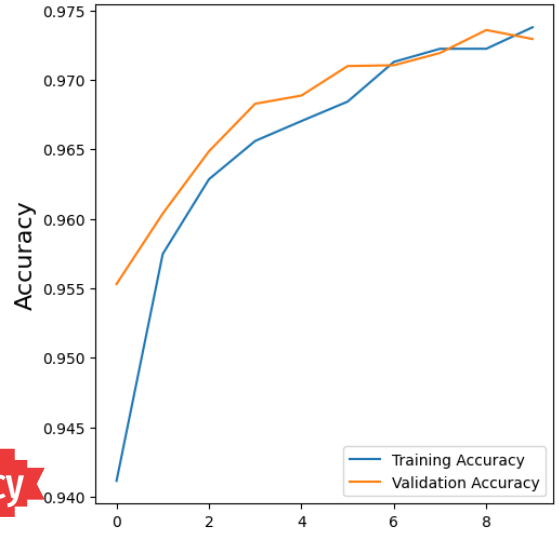
Model training

```
Epoch 9/10
1061/1061 [==============] - 192s 181ms/step - loss: 0.0726 - accuracy: 0.9722 - val_loss: 0.0721 - val_accuracy: 0.9736
Epoch 00009: val_loss improved from 0.07551 to 0.07215, saving model to ./data set/data/premodel.h5

Epoch 10/10
1061/1061 [==============] - 192s 181ms/step - loss: 0.0705 - accuracy: 0.9738 - val_loss: 0.0749 - val_accuracy: 0.9729
Epoch 00010: val_loss did not improve from 0.07215
```



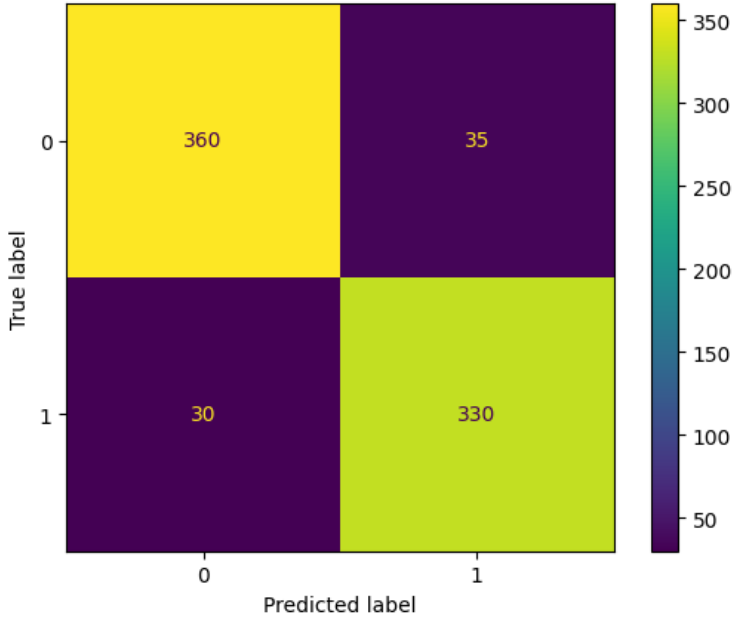




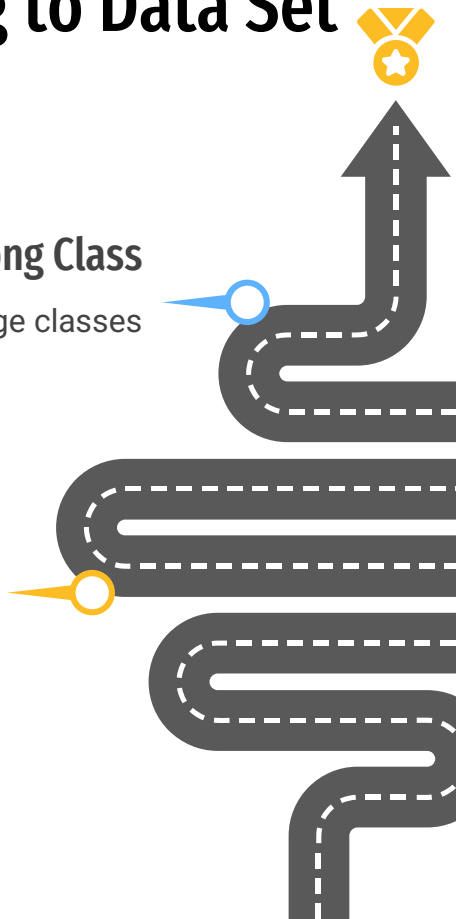
Model - InceptionV3

Model Evaluation

	precision	recall	f1-score	support
Closed	0.92	0.91	0.92	395
Open	0.90	0.92	0.91	360
accuracy			0.91	755
macro avg	0.91	0.91	0.91	755
weighted avg	0.91	0.91	0.91	755



Merge & cleaning to Data Set



Wrong Class

Some images are in wronge classes

Bad Models

Bad Models in Integration with cv2

Duplication

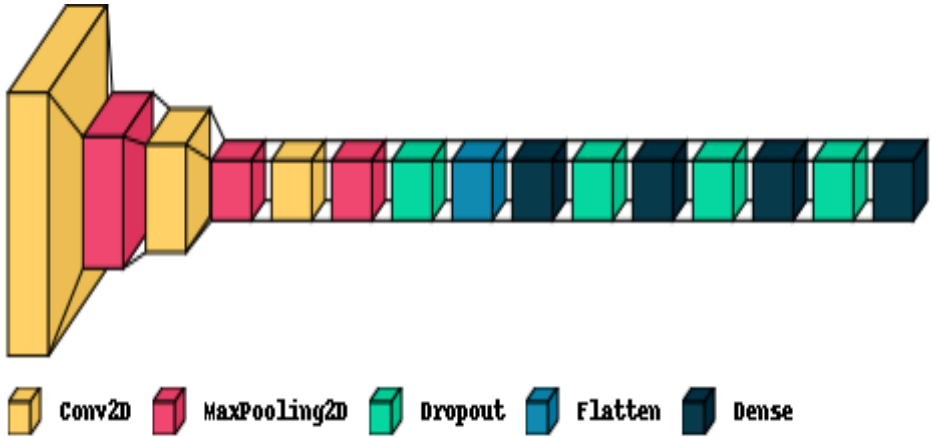
Duplication of old Images

Quality

Low Quality of old Images

Model - CNN

Model Architecture



Layer (type)	Output Shape	Param #
conv2d_97 (Conv2D)	(None, 22, 22, 32)	320
max_pooling2d_5 (MaxPooling2	(None, 11, 11, 32)	0
conv2d_98 (Conv2D)	(None, 9, 9, 32)	9248
max_pooling2d_6 (MaxPooling2	(None, 4, 4, 32)	0
conv2d_99 (Conv2D)	(None, 2, 2, 64)	18496
max_pooling2d_7 (MaxPooling2	(None, 1, 1, 64)	0
dropout_5 (Dropout)	(None, 1, 1, 64)	0
flatten_2 (Flatten)	(None, 64)	0
dense_9 (Dense)	(None, 128)	8320
dropout_6 (Dropout)	(None, 128)	0
dense_10 (Dense)	(None, 64)	8256
dropout_7 (Dropout)	(None, 64)	0
dense_11 (Dense)	(None, 32)	2080
dropout_8 (Dropout)	(None, 32)	0
dense_12 (Dense)	(None, 2)	66

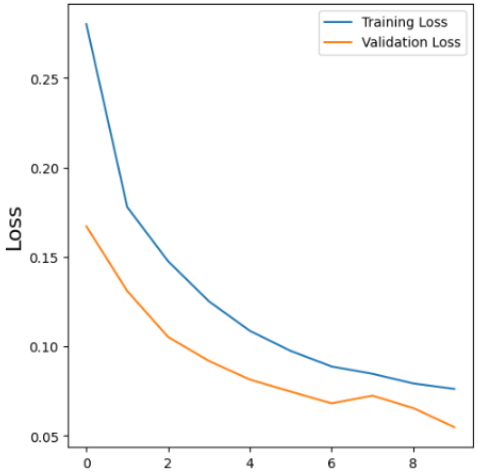
Total params: 46,786
Trainable params: 46,786
Non-trainable params: 0

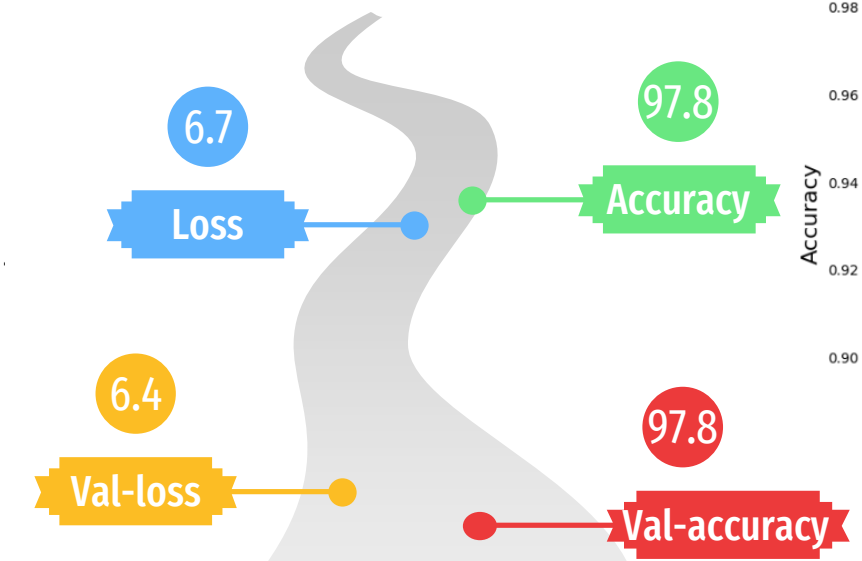
Model-CNN

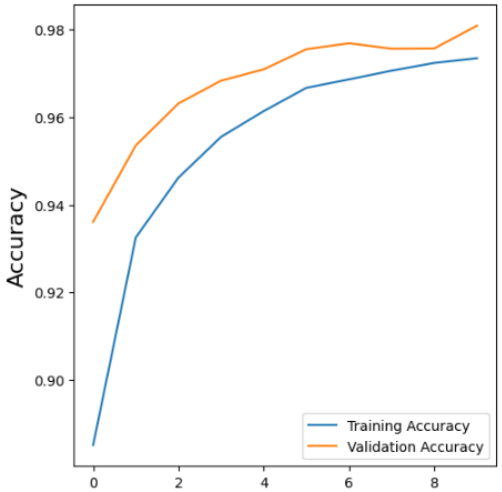
Model training

```
Epoch 9/10
1737/1737 [==============] - 107s 61ms/step - loss: 0.0702 - accuracy: 0.9769 - val_loss: 0.0760 - val_accuracy: 0.9748
Epoch 00009: val_loss did not improve from 0.06524

Epoch 10/10
1737/1737 [==============] - 109s 63ms/step - loss: 0.0670 - accuracy: 0.9779 - val_loss: 0.0642 - val_accuracy: 0.9782
Epoch 00010: val_loss improved from 0.06524 to 0.06421, saving model to ./data set/data/cnnfinal.h5
```



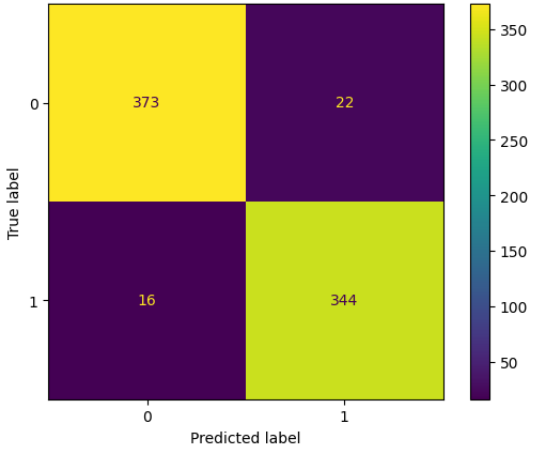




Model - CNN

Model Evaluation

	precision	recall	f1-score	support
Closed	0.97	0.96	0.97	396
Open	0.96	0.97	0.96	360
accuracy			0.96	756
macro avg	0.96	0.96	0.96	756
weighted avg	0.96	0.96	0.96	756



Cascade Classifier

To detect faces and eyes
From frame

Preprocessing

We applied prerocessing
For eye like resize and to
grayscal and etc.

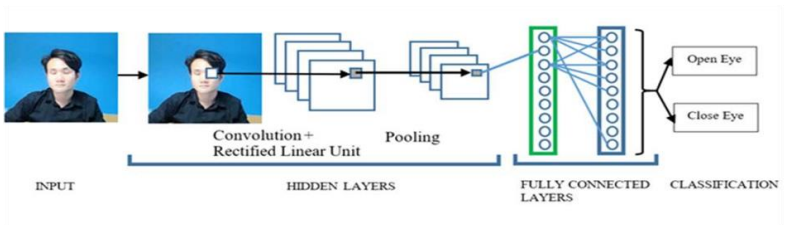
Integration with cv2

Alarm

If score more than 5 .the
alarm will begin make
sound

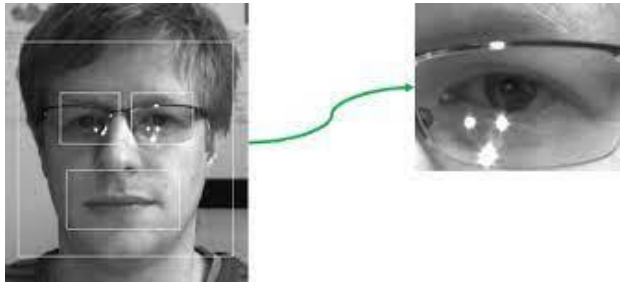
Score

If eye was opened .score
Will increase and vice
versa

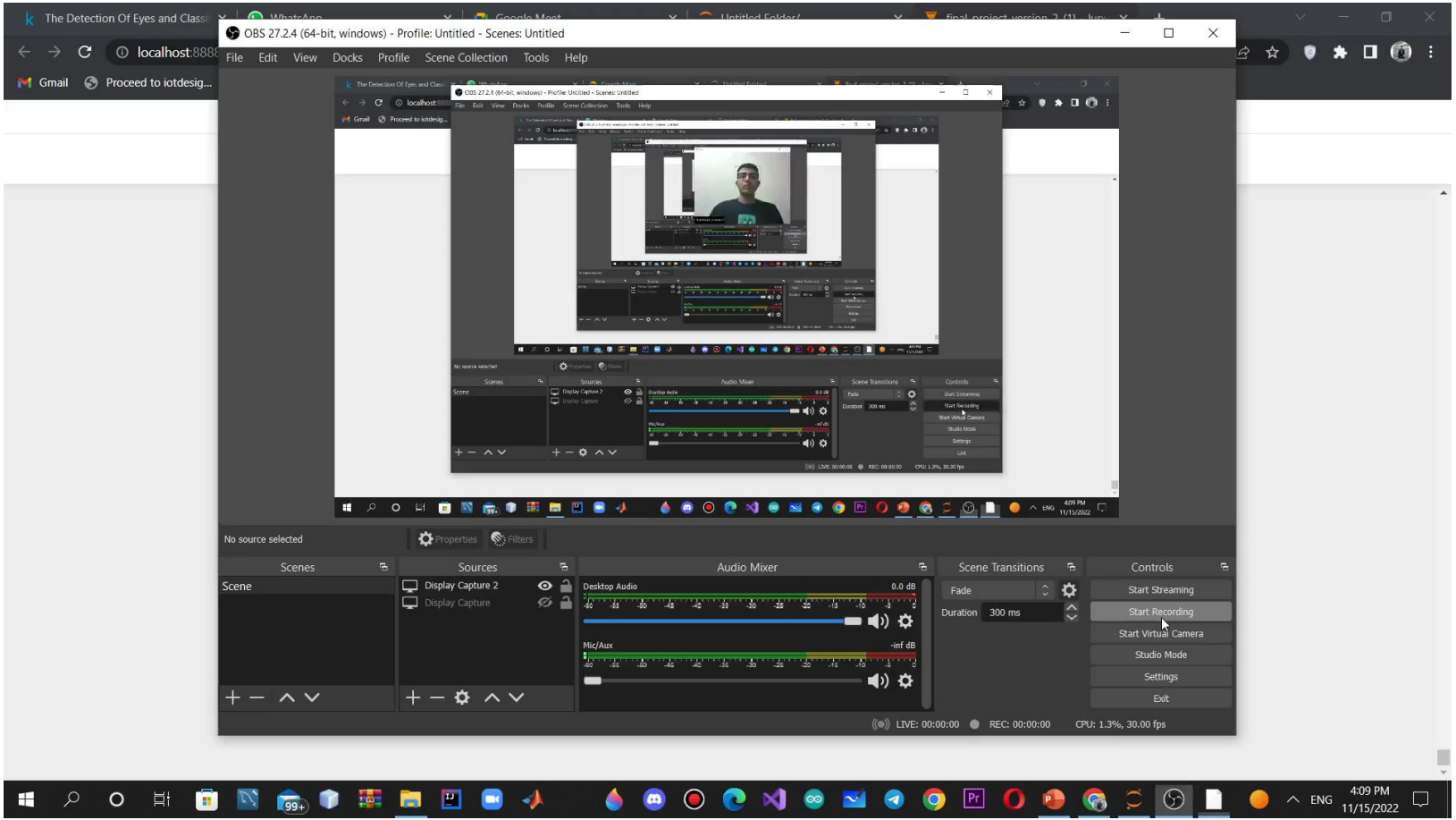


To model

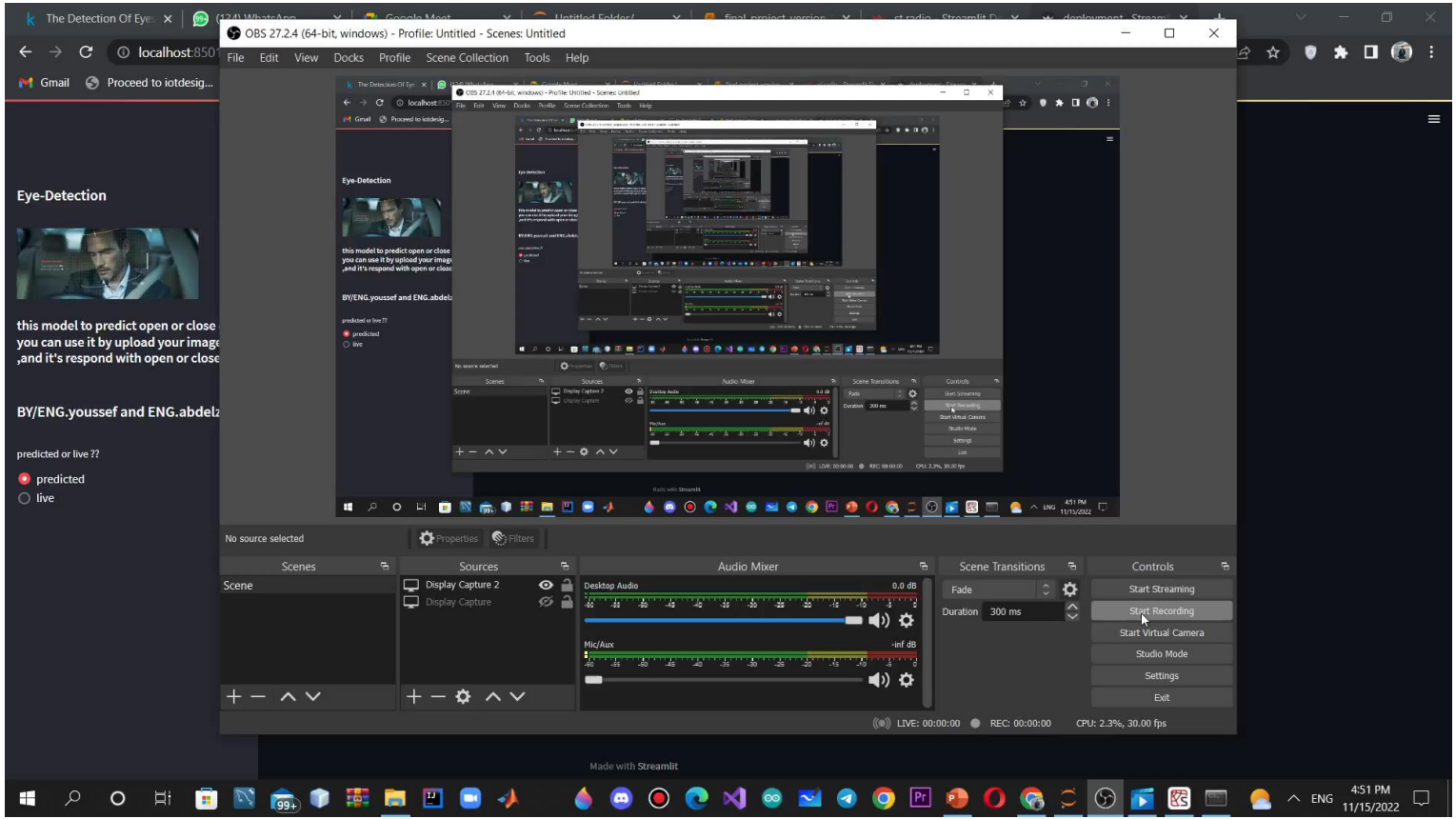
The model predicted
if eye was opend or
closed



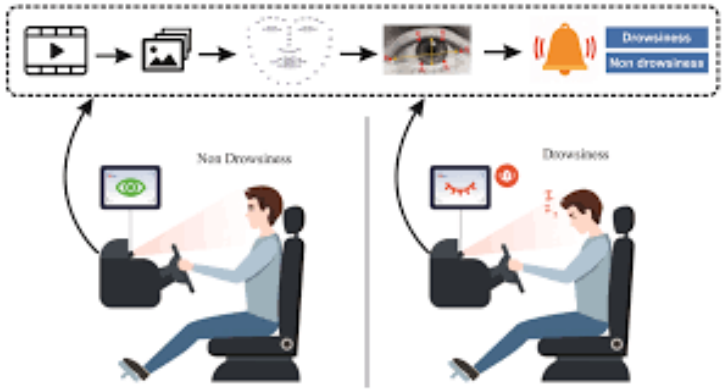
Video Demo


OBS 27.2.4 (64-bit, windows) - Profile: Untitled - Scenes: Untitled
File Edit View Docks Profile Scene Collection Tools Help
No source selected
Properties
Filters
Scenes
Scene
Sources
Display Capture 2
Display Capture
Audio Mixer
Desktop Audio
0.0 dB
Mic/Aux
-inf dB
Scene Transitions
Fade
Duration
300 ms
Controls
Start Streaming
Start Recording
Start Virtual Camera
Studio Mode
Settings
Exit
LIVE: 00:00:00
REC: 00:00:00
CPU: 1.3%, 30.00 fps

Deployment with streamlit



Recomendtion



Ready to Ship In stock Fast dispatch

Car Anti Sleep Driver Monitoring Camera Car Alarms For Driver Long Time Driving Fatigue Security Warning

1 - 99 Pieces	100 - 499 Pieces	>= 500 Pieces
$52.00	$50.00	$48.00

Contact supplier | Chat now | View product details

Most of the traditional methods for drowsiness detection are based on behavioral factors, while some require expensive sensors and devices to measure sleepiness and we produced good model With accuracy 97% And made a protype this case study and we can to made a product and will sell It in Egyptian market with low prices

Team

Abdelaziz Nabil
Faculty of Engineering
At Alexandria University

Youssef walid
Faculty of computers
At Benha University

Supervised By :

Dr. Shady Nagy
Eng. Ola

SAMSUNG

Makers EGYPT

Together for Tomorrow!

Enabling People

Education for Future Generations