DeepVisionTech.AI

Realize your AI vision with us

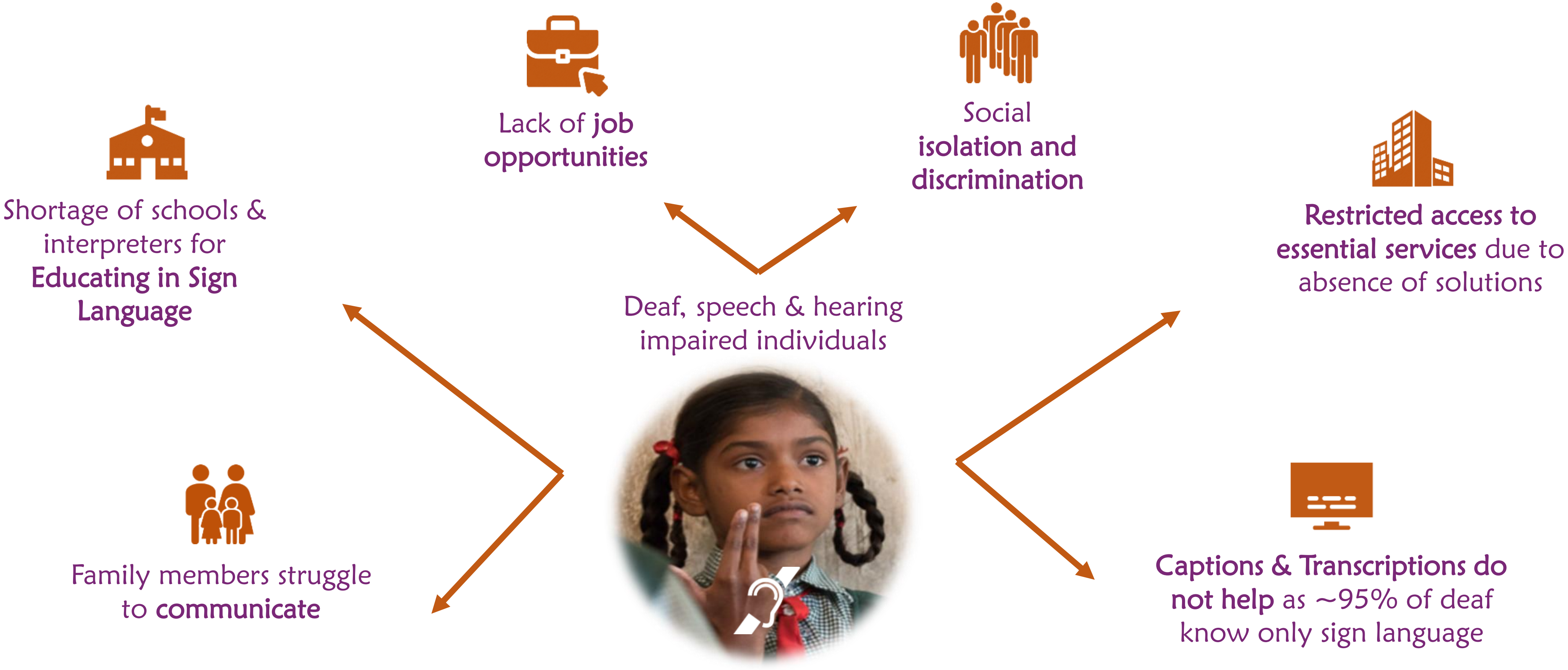
We **Empower People with Disabilities** to live independently using innovative solutions based on AI

Let's Talk Sign®

Interpreter Anytime, Anywhere.

Will Empower millions of **Deaf, Speech & Hearing impaired people to 'Live Independently'** and will enable **Businesses to be Inclusive.**

‘Hearing Impairment’ impacts 466m* in the world and 80m# in India



Shortage of schools & interpreters for **Educating in Sign Language**

Lack of **job opportunities**

Social **isolation and discrimination**

Restricted access to essential services due to absence of solutions

Deaf, speech & hearing impaired individuals

Family members struggle to **communicate**

Captions & Transcriptions do not help as ~95% of deaf know only sign language

‘Communication’ barrier is just one of the many challenges…

…there is hardly any solution

* WHO and # Based on estimation that 63m in 2016 IJO

5 variants to address all key challenges of deaf, speech & hearing impaired people

Communication

For **Personal Communications** (on phone)

1

Education in Sign Language

For **Interpreting** Text Books by deaf students to learn in ISL

2

Digital Content Accessibility

For **Interpreting Digital Contents on Websites (Accessibility for deaf)**

3

Inclusion at Workplace

For **Workplace Communications** (on phone & PC)

4

Customer Support Accessibility

For **enabling Chatbots to communicate in Sign Language**

5

Access to essential Services

For **Communication in public offices** (on kiosks)

1

One variant of Let'sTalkSign to solve each challenge

Lets discuss about what is AI

AI for Assistive Technology Accessibility

DeepVisionTech.AI
Realize your AI vision with us

AI can 'SEE' and 'READ'

Reason1: AI can 'see' – Computer Vision

- **Recognizing objects:** solution for blind & visually impaired.

  Ex: SmartCane by IIT Delhi

- **Action recognition:** solution for deaf for interpreting sign language.

  Ex: Let'sTalkSign

Hello my name Arya and me Interpret Speech and Text to Sign Language to help Deaf and Hard-of-hearing people

hello my name is Arya and I Interpret speech and text to Sign Language to help deaf and Hard-of-hearing people

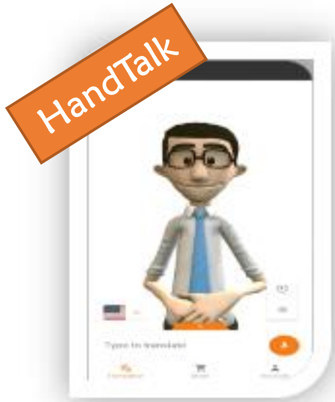
Reason2: AI can 'read' – Natural Language Understanding

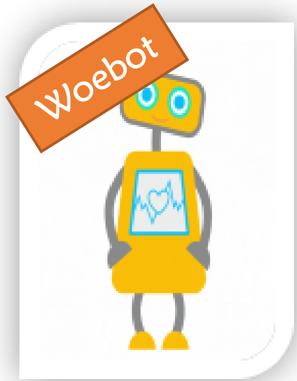
- **Language Translation:** any language to Sign Language.

  Ex: HandTalk (Brazilian Sign Language), Let'sTalkSign

- **Chatbots and auto-responses:** for providing mental health assistance.

  Ex: Woebot does survey based analysis





AI can ‘HEAR’ and ‘TALK’

Reason3: AI can ‘hear’ – Speech Recognition

- **Voice controlled devices:** for people with physical disabilities.

  Ex: Smart Home devices like Echo or Home

- **Audio transcription:** solution for hearing impaired.

  Ex: SpeakLiz, Ava

...pp transcribes ...ces around you and even identifies your sound surroundings

Male voice

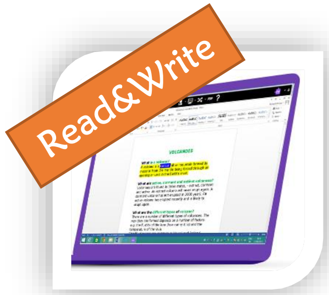
Reason4: AI can ‘talk’ – Speech Synthesis

- **Text-to-Speech:** solution for dyslexia patients.

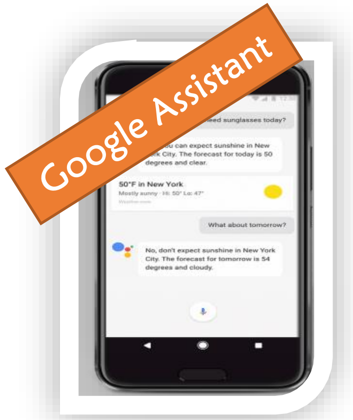
  Ex: Read&Write

- **Voice synthesis:** solution for people who lost their voice.

  Ex: Avaz



AI can 'CONVERSE' and 'DO TASKS'



eed sunglasses today?

ou can expect sunshine in New rk City. The forecast for today is 50 degrees and clear.

50°F in New York
Mostly sunny · Hi: 50° Lo: 47°

What about tomorrow?

No, don't expect sunshine in New York City. The forecast for tomorrow is 54 degrees and cloudy.

Reason5: AI can 'make conversations' – Conversational AI

- **Virtual Assistants:** solution for people with disabilities & blind.

  Ex: Google Assistant, MS Cortana

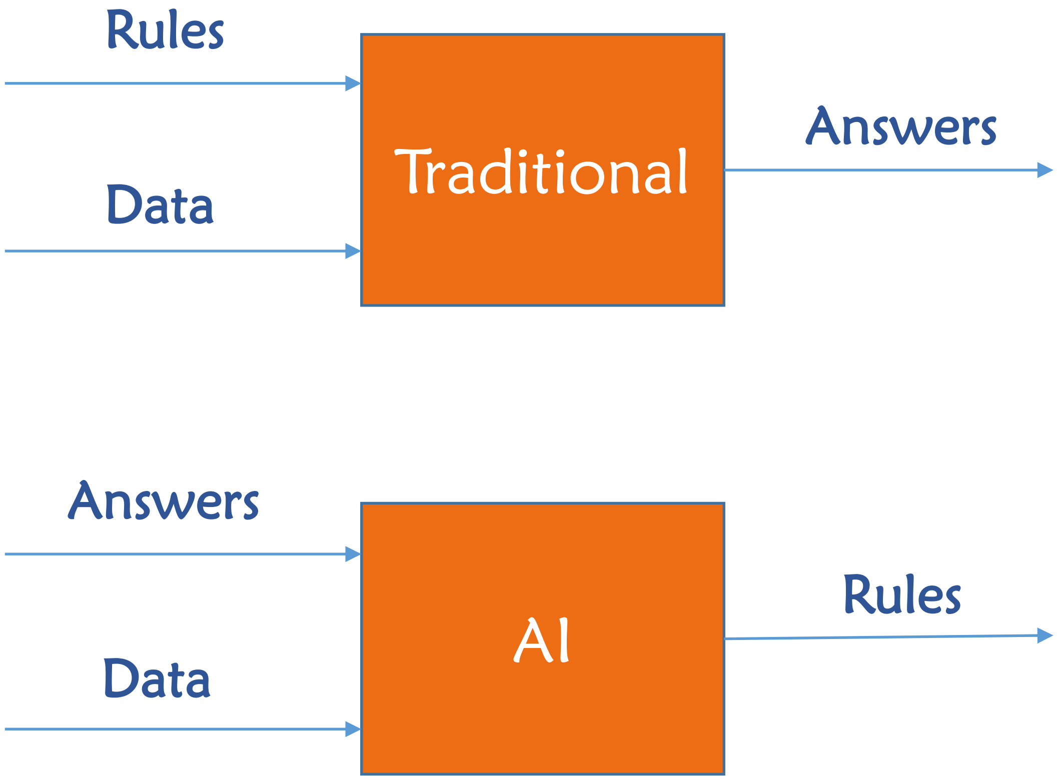
- **Chatbots:** for personalized & contextual conversations for elderly & people with mental illness, autism, etc.

Reason6: AI can 'do tasks' – Robotics

- **Smart exoskeleton, Smart prosthetics:** for people with physical disabilities to do physical activities.

  Ex: TWIICE by Switzerland's Technical University

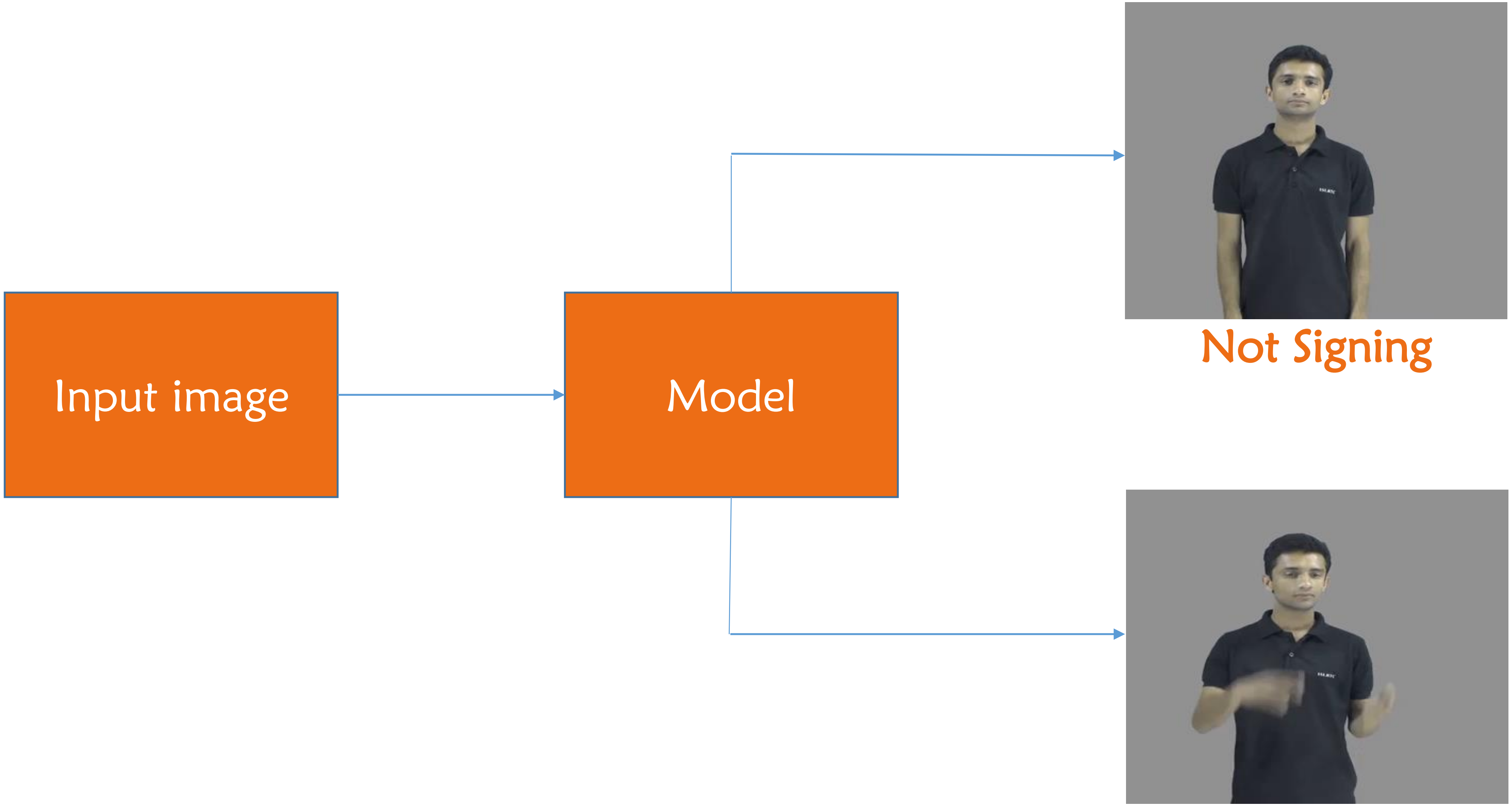
Traditional programming vs AI programming



Traditional	AI
Developing computing functionality by writing instructions using programming language.	AI is a module that can self-learn and improve. Functionality of AI is formed by training.
Output depends entirely on the algorithms implemented	Output depends on multiple factors like dataset, features chosen, model arch, hyperparameters, training approach, etc.
Well defined algorithms for different functionalities needed	Need to figure-out the best algorithm for model by using appropriate dataset and experimenting with various architectures

Our project:
Model to identify whether a person is ‘signing’ or not

Our Project



Not Signing

Signing

Purpose of this model?

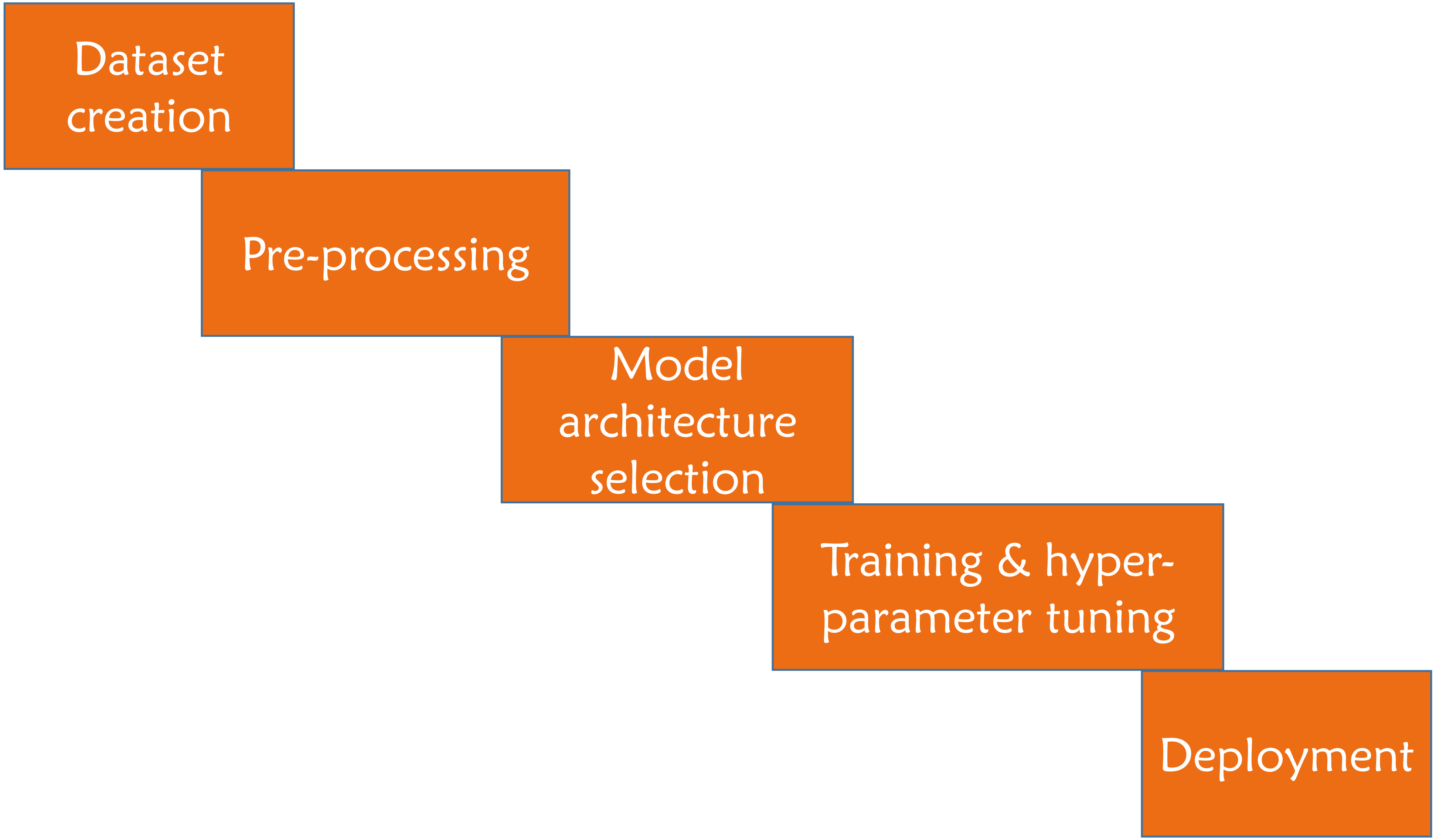
Uses

- in video conferencing apps – helps in detecting if a deaf person in the call is signing or not so that the other participants can be informed about it
- in our communication app – helps decide when to start & stop our sign recognition technique

Advantages

- in video conferencing apps – helps give chance to deaf attendees interject conversations and share their inputs
- in our communication app – helps save unnecessary usage of compute power on devices like smartphones, kiosks, etc.

Steps involved in creating a deep learning model?


Dataset creation

Pre-processing

Model architecture selection

Training & hyper-parameter tuning

Deployment

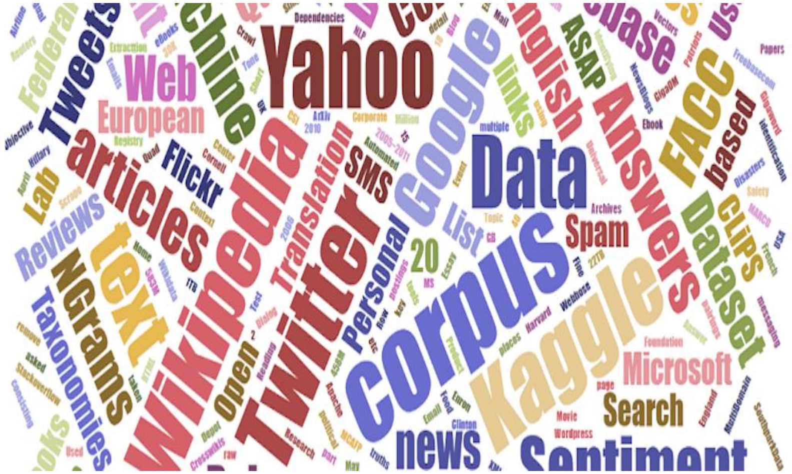
Dataset creation

What is a dataset ?

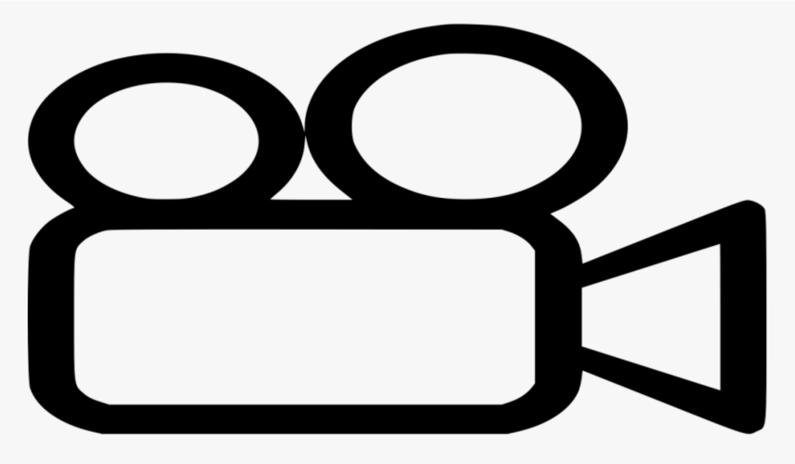
Types of data

Sound

Text

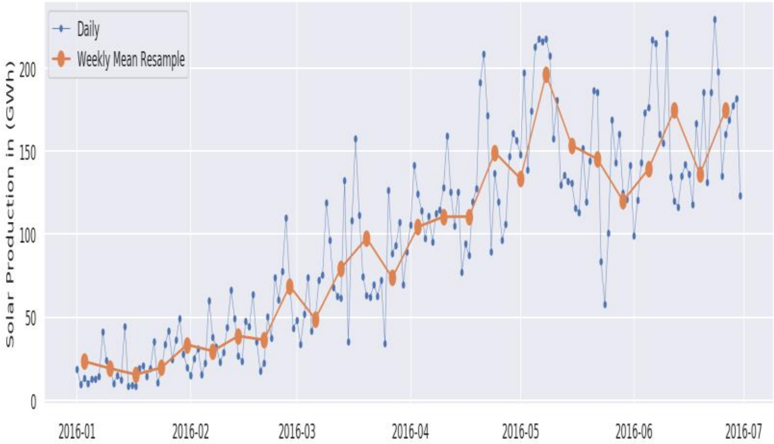


Video

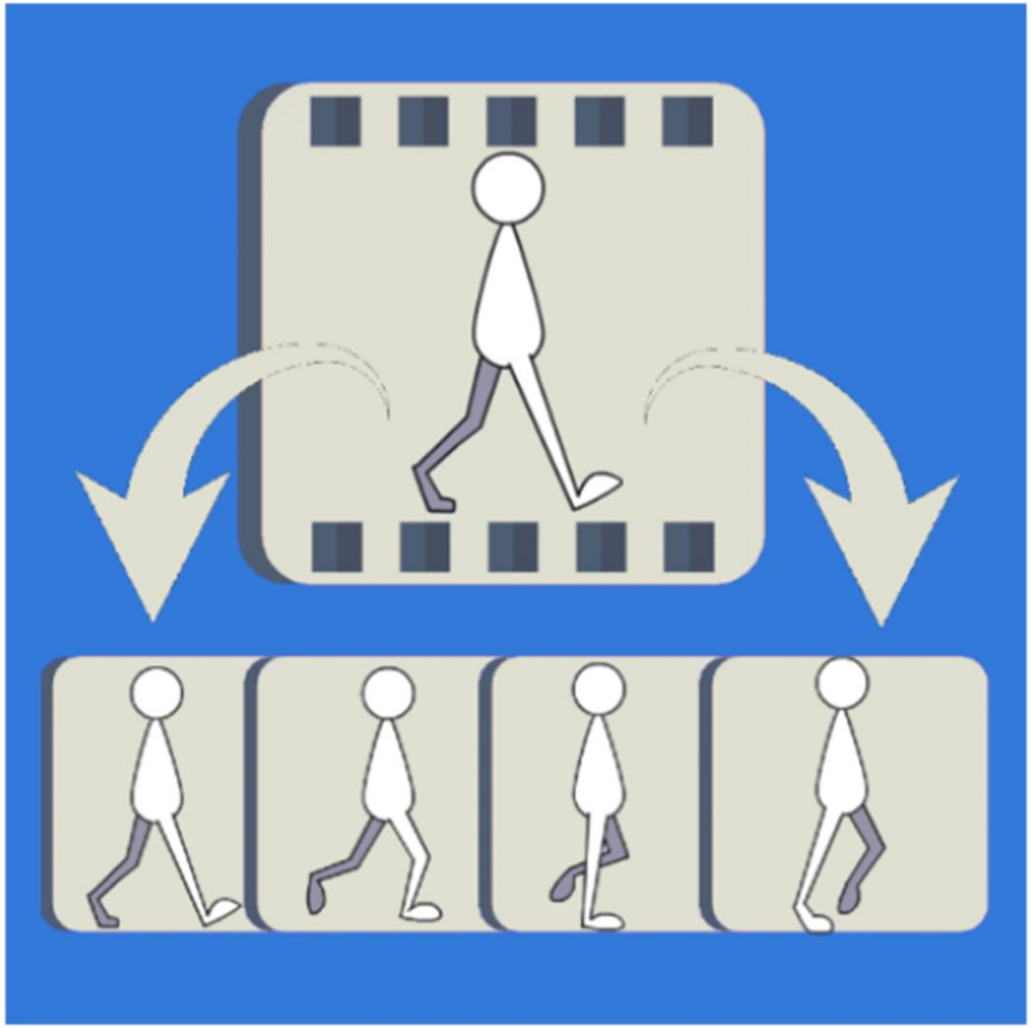


Image

Time series



What composes a video?



- A video is basically a collection of frames stitched together temporally.
- In our case we don’t need the temporal information to predict whether a person is signing or not.
- So we are going to convert the input the video to frames using a tool called as ffmpeg.

Pre-Processing

Libraries required

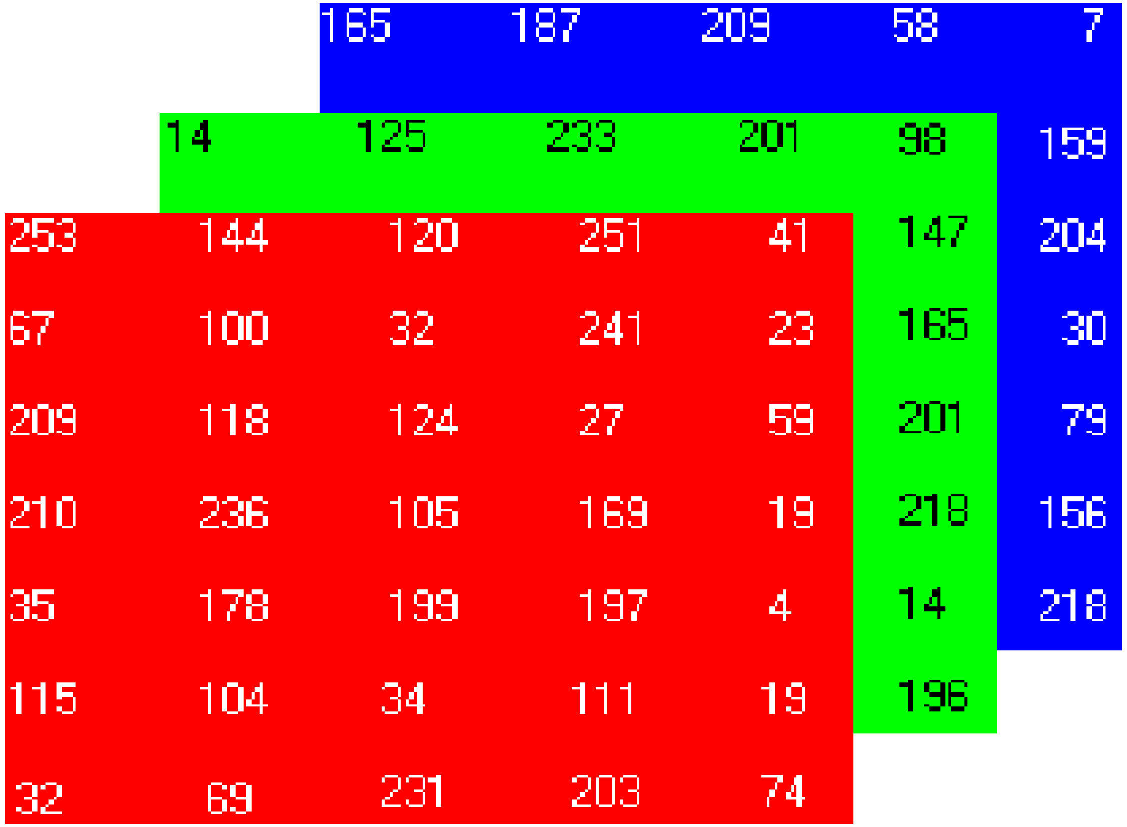
NumPy

OpenCV

OS Module

```
>>> import os
```

Digital image

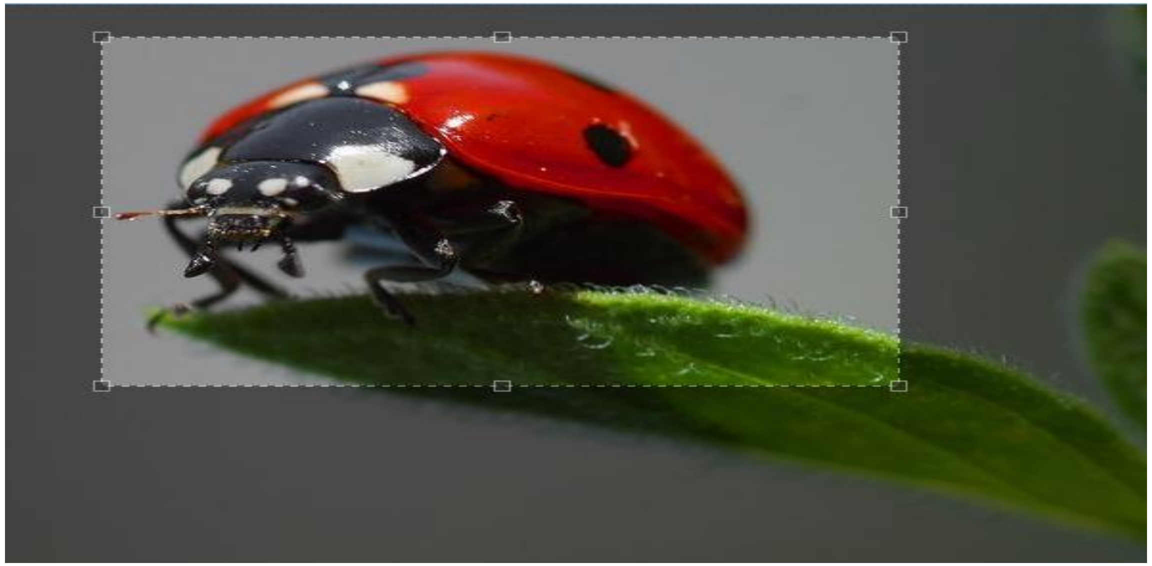


165	187	209	58	7
				159
				204
				30
				79
				156
				218

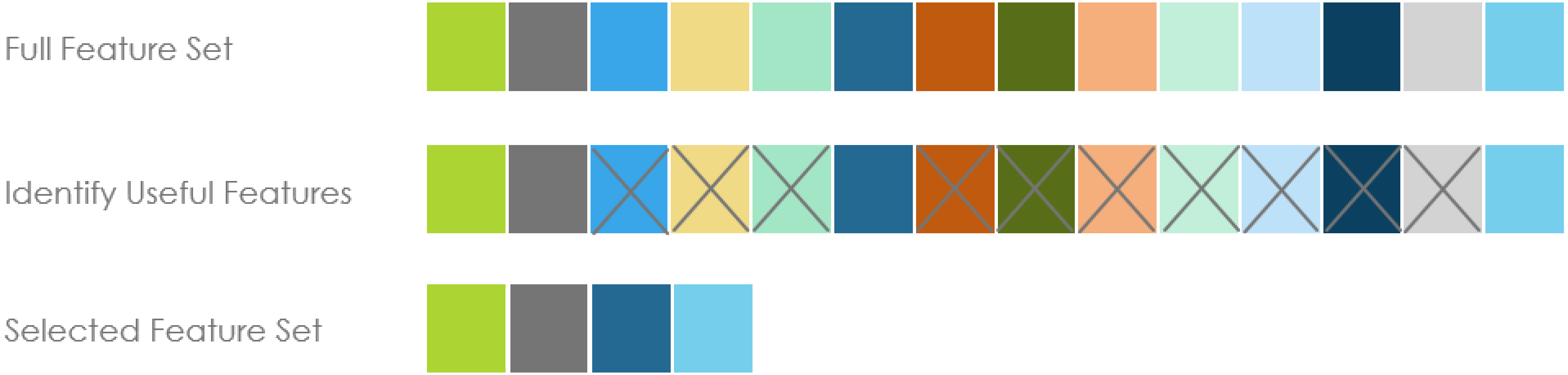
14	125	233	201	98
				147
				165
				201
				218
				14
				196

253	144	120	251	41
67	100	32	241	23
209	118	124	27	59
210	236	105	169	19
35	178	199	197	4
115	104	34	111	19
32	69	231	203	74

Cropping & Resizing



Feature selection

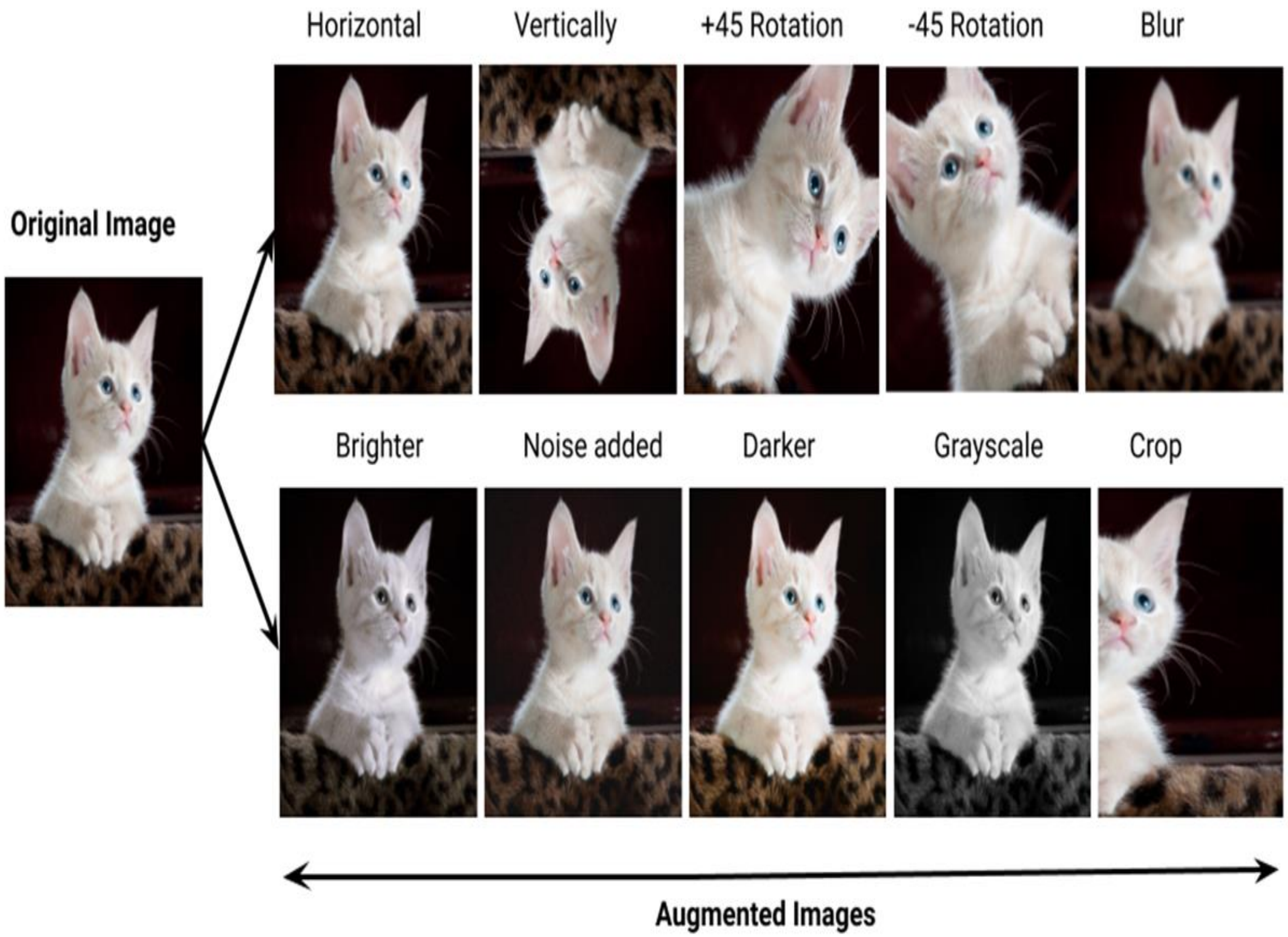


Full Feature Set

Identify Useful Features

Selected Feature Set

Augmentation

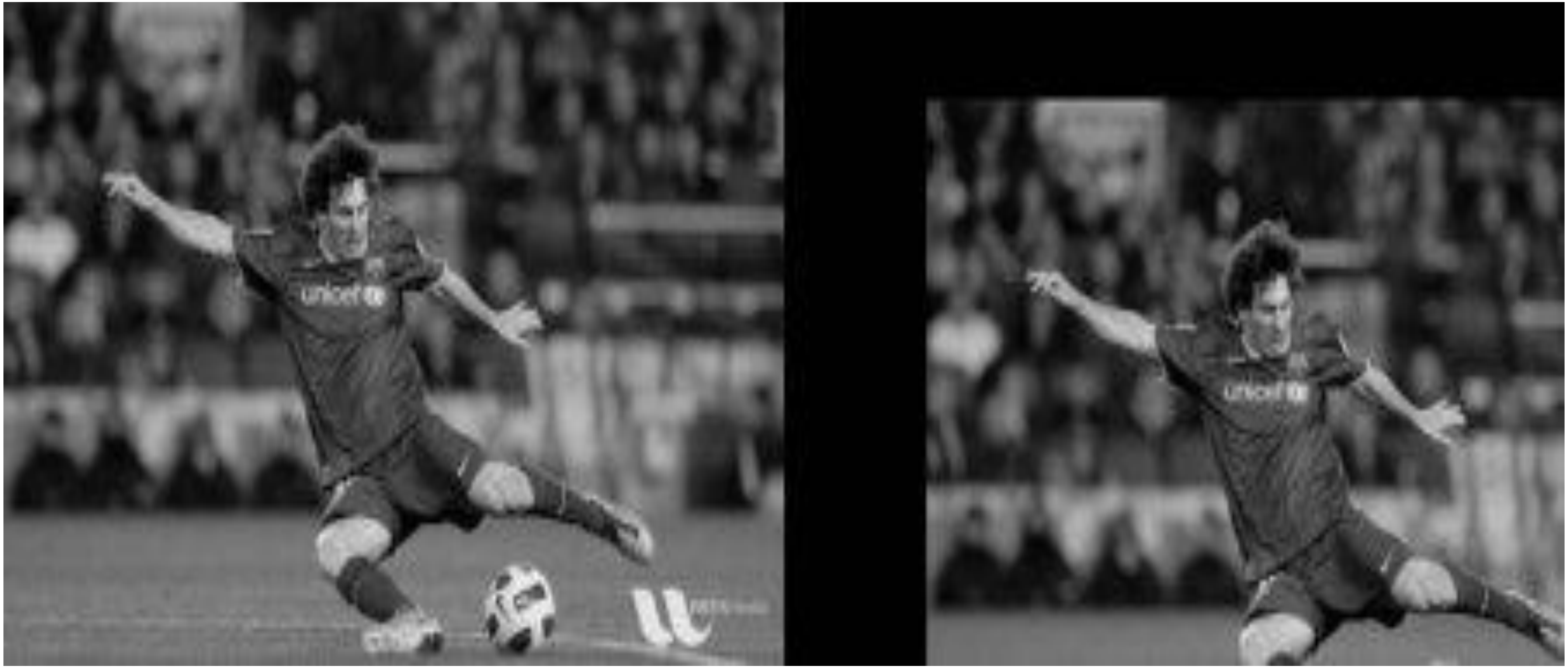


A small intro to open-cv

Some operations that can be done using Open - CV

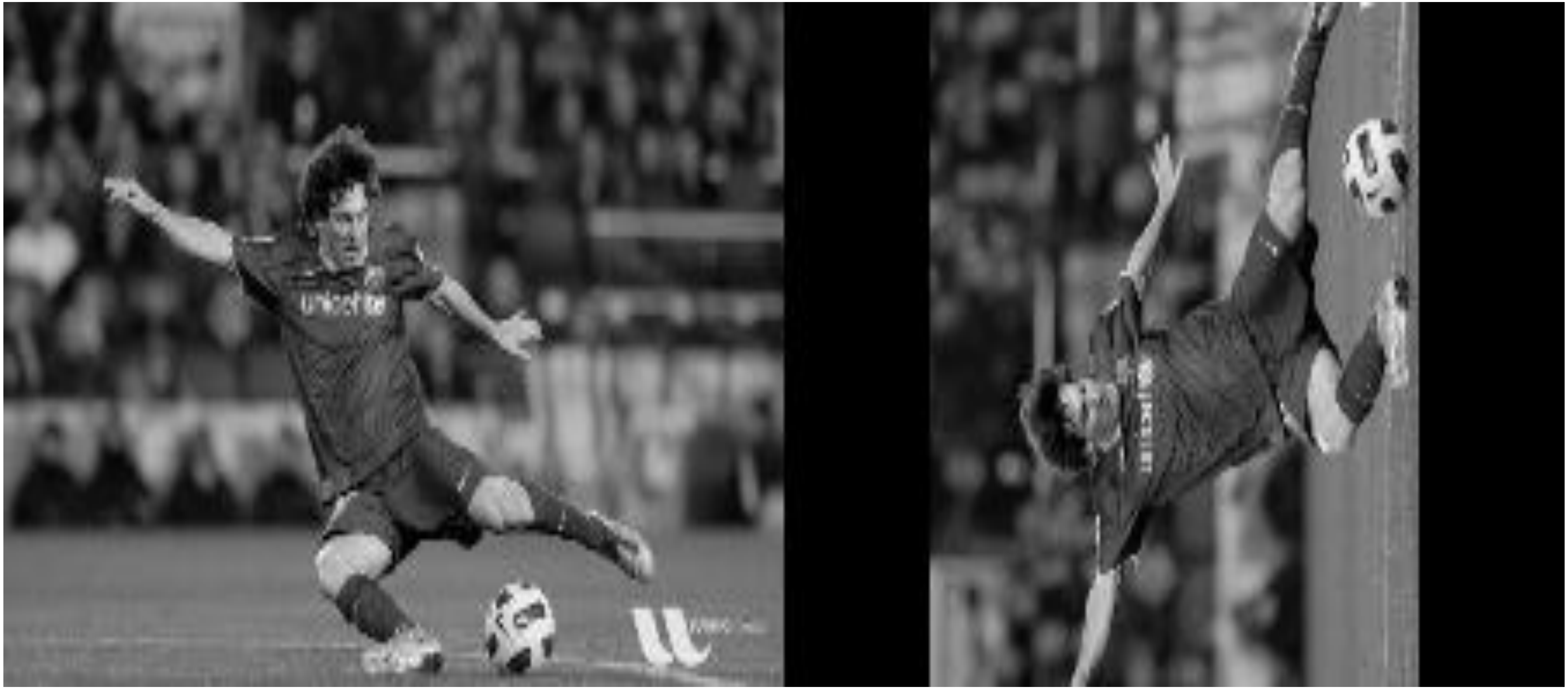
- Image translation
- Rotation
- Affine transformation
- Image Blurring
- Image gradients

Image translations



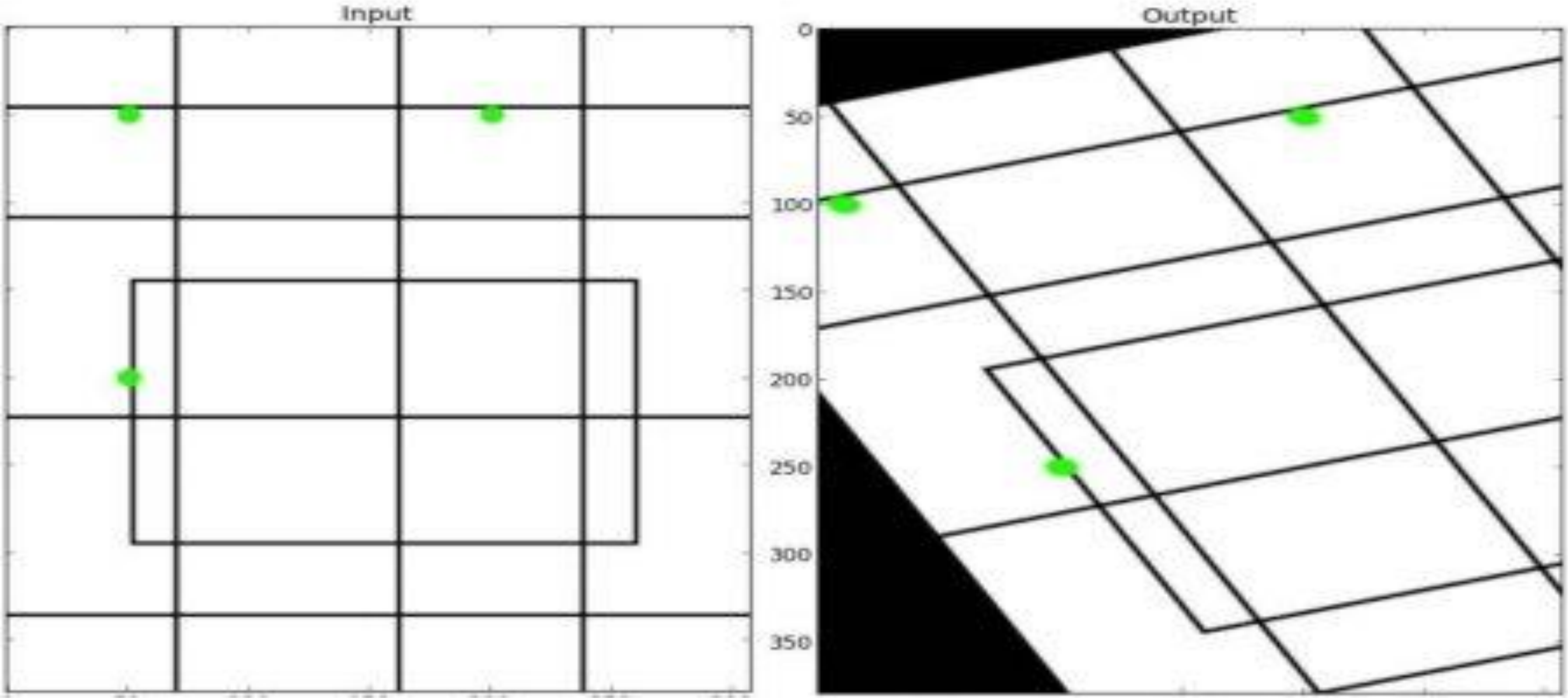
https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html#:~:text=interpolation%20%3D%20cv.INTER_CUBIC)-,Translation,-Translation%20is%20the

Rotation



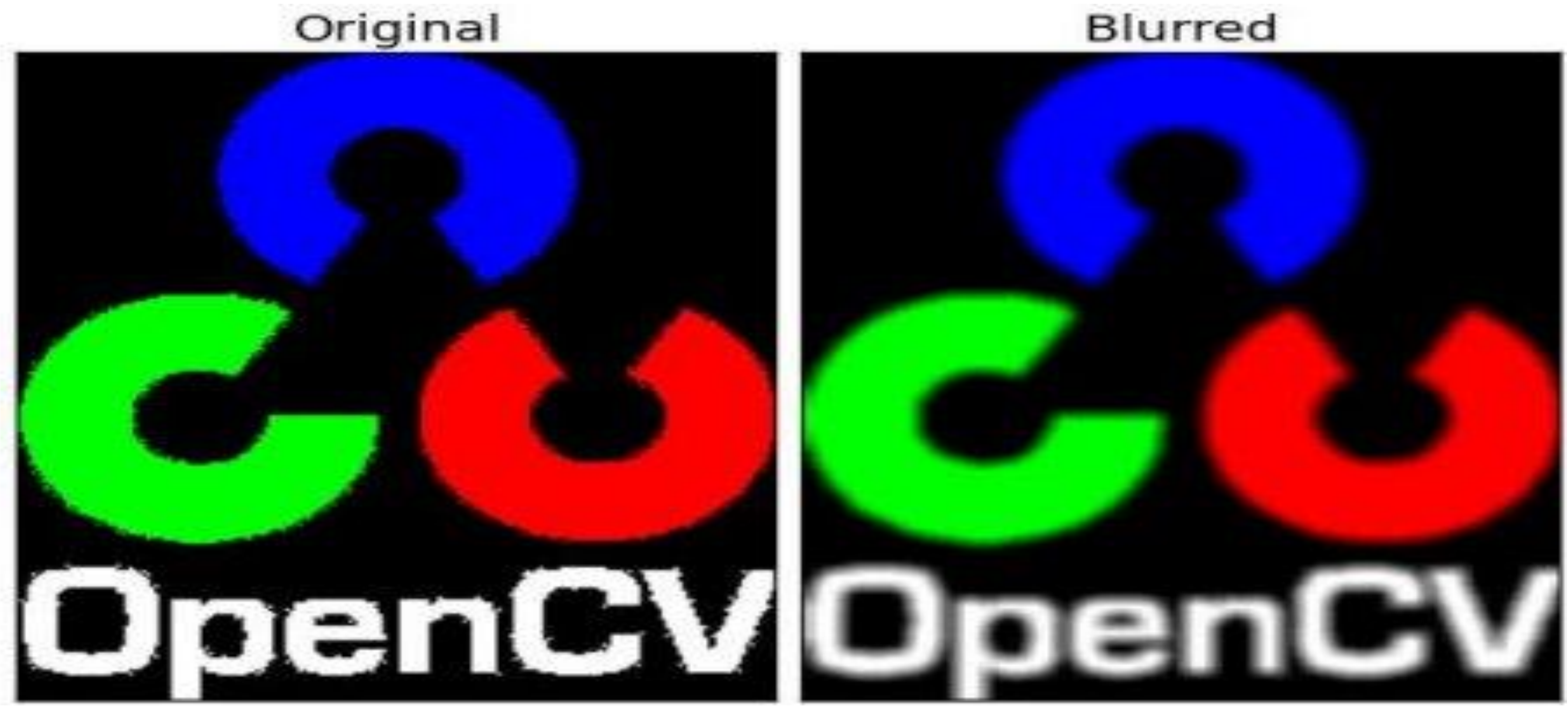
https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html#:~:text=image-,Rotation,-Rotation%20of%20an

Affine transformation



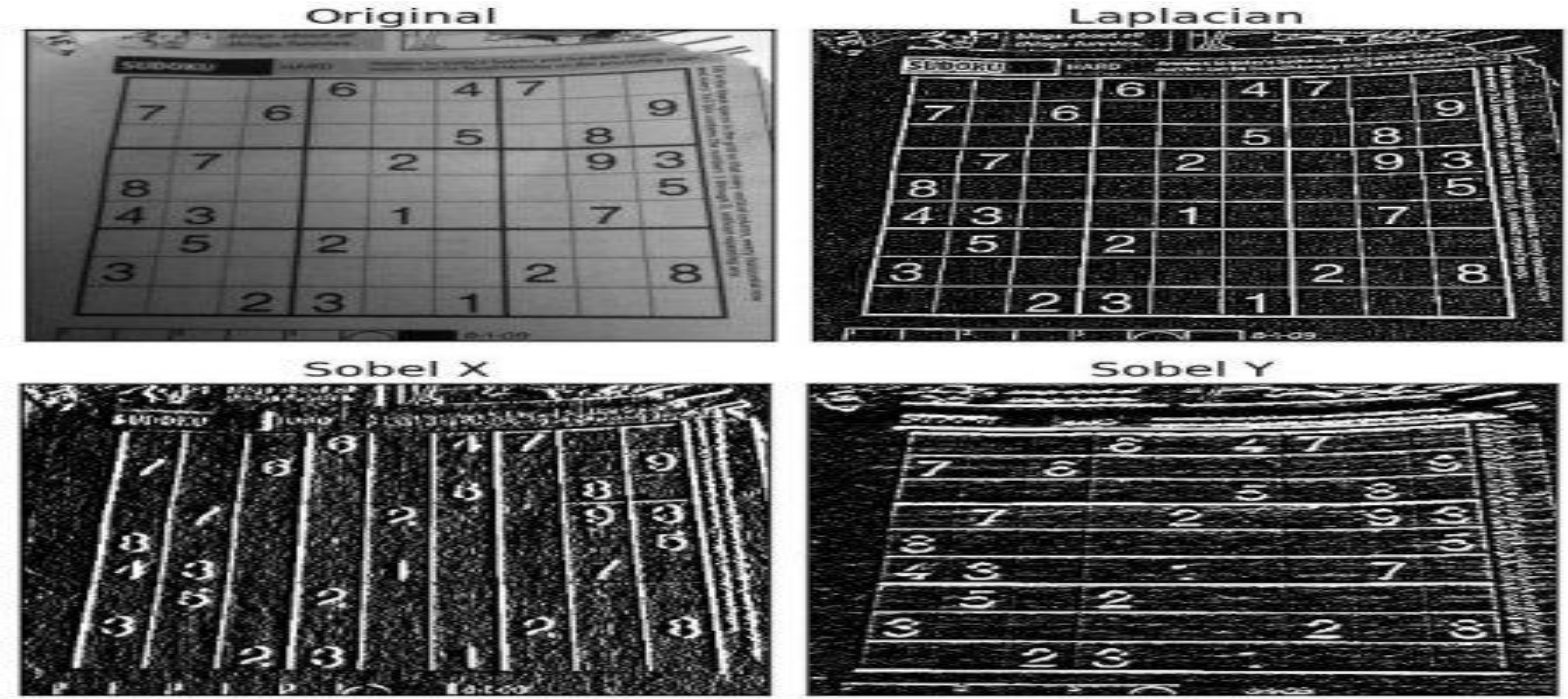
https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html#:~:text=image-,Affine%20Transformation,-In%20affine%20transformation

Image blurring



https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html#:~:text=image-,Image%20Blurring,-(Image%20Smoothing)

Image gradients

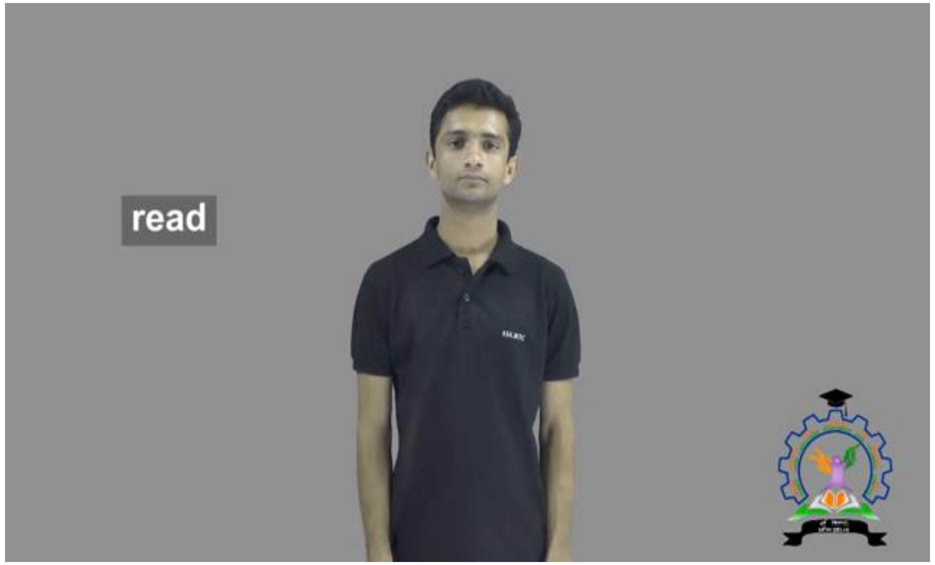


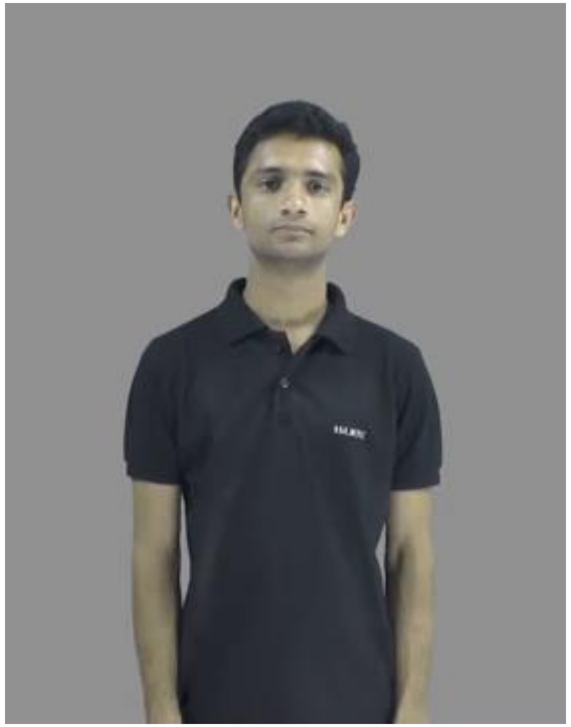
https://docs.opencv.org/4.x/d5/d0f/tutorial_py_gradients.html#:~:text=cv.Laplacian()%20etc-,Theory,-OpenCV%20provides%20three

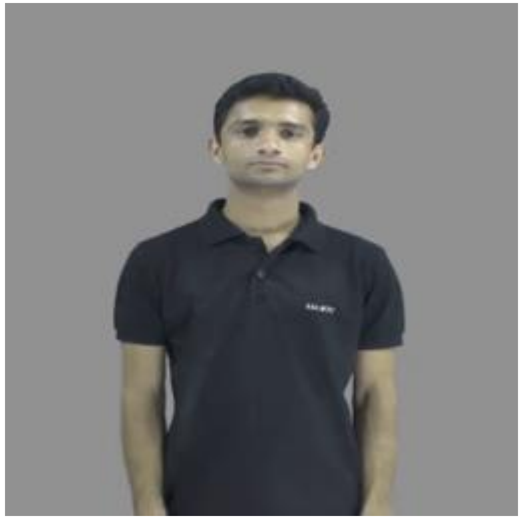
The techniques we are going to use

Convert video to frames

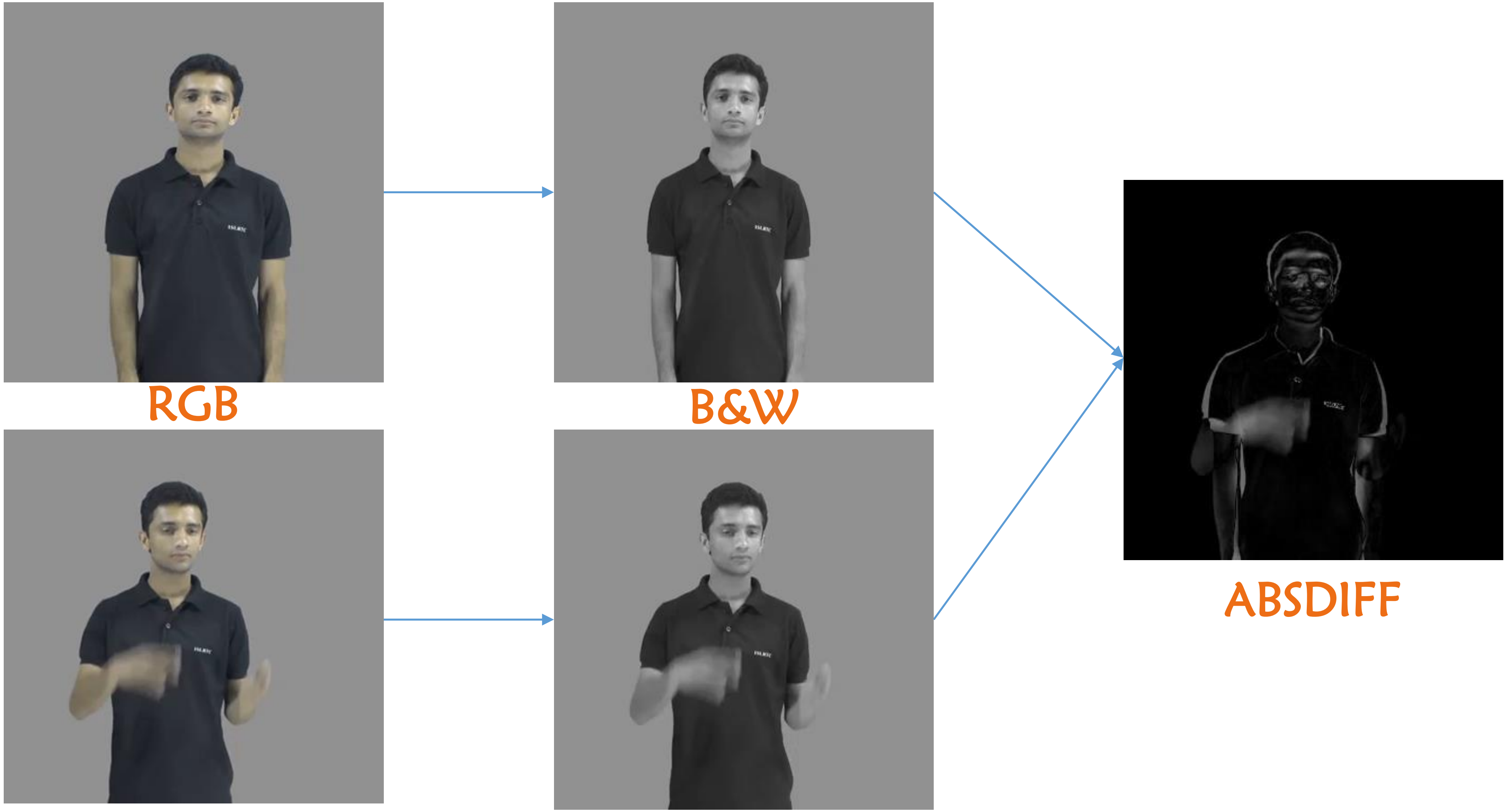
Cropping and resizing (Pre-processing)







Feature selection (Pre-processing)



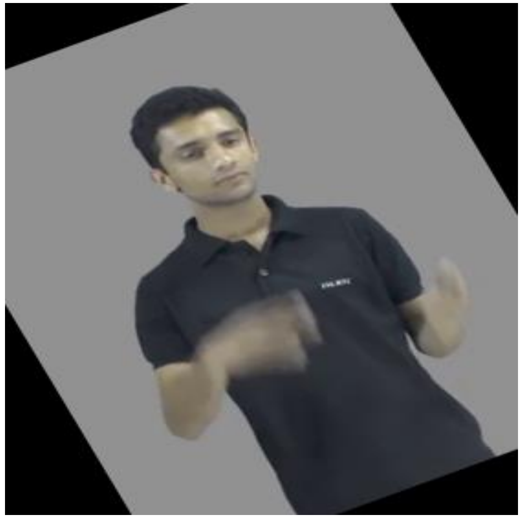
RGB

B&W

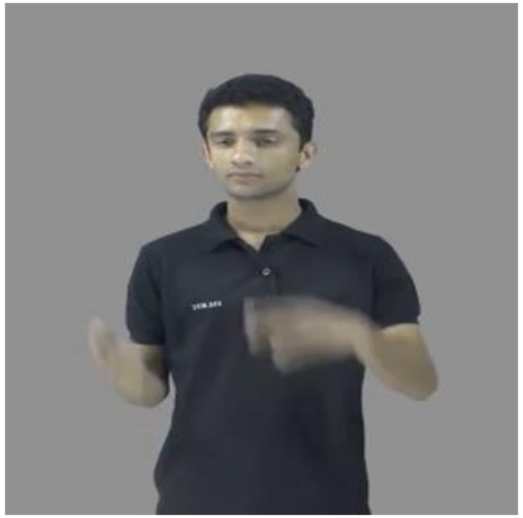
ABSDIFF

Augmentation

Input



Rotate

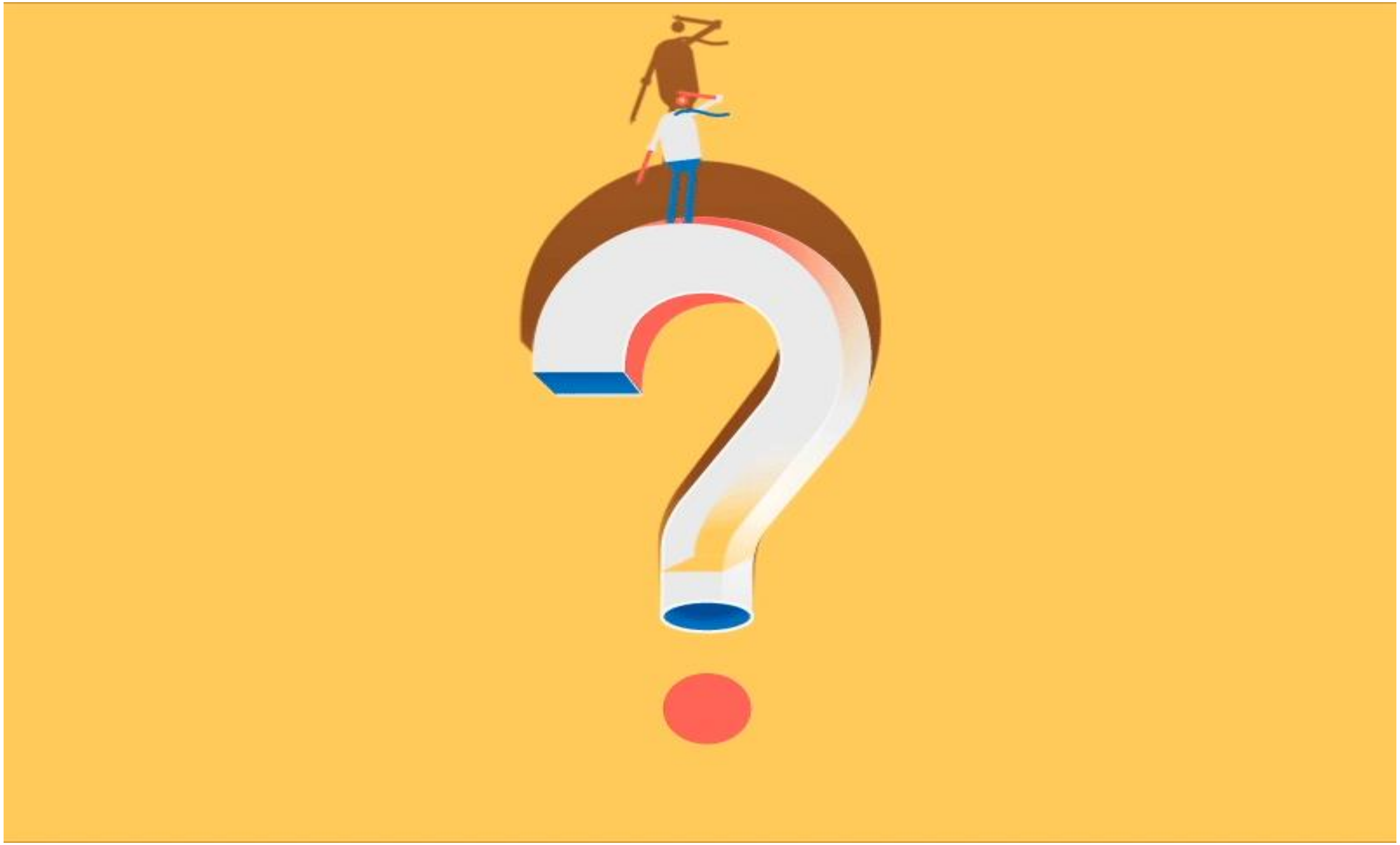


Flip

TROUBLE SHOOTING & TIPS

- In google colab you cannot display a image as you normally do in opencv as a work around you can use a patch in colab which you can import to display a image.
- You can also attach your google drive to colab and use the dataset from there.
- In opencv there is no build in function to rotate a image based on the angle but you can use numpy to build a function for it. Like this you can manipulate the image in multiple ways with your own functions build using numpy.

Any doubts



References

OPEN-CV

- https://towardsdatascience.com/complete-image-augmentation-in-opencv-31a6b02694f5
- https://docs.opencv.org/4.x/d9/df8/tutorial_root.html

NUMPY

- https://towardsdatascience.com/data-augmentation-compilation-with-python-and-opencv-b76b1cd500e0

SCRIPT REFERENCE

- https://colab.research.google.com/drive/1ekQu5XLvmrvpcFYiH6wVPDfB-O3p9GTF?usp=sharing