

| Last Updated By: | Gurram Vamshi Kumar |
| --- | --- |
| Date: | 13-06-2023 |



**Chapter 1: Introduction**

This document is aimed at guiding through the code block / Class (ImageOrientationCorrection)that will automatically detect the orientation of any objects in an image and rotate the image with respect to that image orientation in opposite direction, making the orientation of the object to the desired one(like 90 degrees in this case and the angle is measure with respect to it).

Key words: Orientation,Eigenvectors,PCA

Sometimes we may miss some features of an object or an image due to its orientation. That time you may need to rotate the image according to any object to extract relevant features from that image. It also helps in many aspects in traditional computer vision techniques.

To rotate the image with the desired orientation or the orientation of a particular object in an image, firstly we need to know the orientation of the respective object. In this python class we have introduced an efficient computer vision technique to predict/estimate the orientation of the desired object. The techniques worked pretty well with a tolerance less than or equal to +/- 0.8 degrees. We will see the detailed explanation of technique in chapter 2. Now we can see some of the keywords and their definitions.

## [**Orientation:**](https://en.wikipedia.org/wiki/Orientation_(geometry))

In geometry the orientation of an object such as aline,plane, rigid body is part of the description of how it is placed in the space it occupies. More specifically, it refers to the imaginary rotation that is needed to move the object from a reference placement to its current placement.

## [**Eigenvectors:**](https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors)

Geometrically, eigenvectors can be thought of as directions in which a linear transformation does not change the length of a vector. For example, if a linear transformation is a rotation, then the eigenvectors of the transformation are the directions in which the rotation does not change the length of a vector. So we can use the two principal components of a point to give the principle axes of that object/point at that place.



## **Pr**[**incipal Component Analysis (PCA) :**](https://docs.opencv.org/4.x/d1/dee/tutorial_introduction_to_pca.html)

Principal Component Analysis (PCA) is a statistical procedure that extracts the most important features of a dataset. if you know the position of a point along the direction of data you have more information about the point than if you only knew where it was on its feature axes. So that will help in this problem so if we draw two eigenvectors at any point we will have two vectors in the direction of the more dense data points(stable data points) and other in the direction in which data varies.So the angle between those eigenvectors will give us the orientation of that object.

**Chapter 2: Methodology and Usage**

The given Class ImageOrientationCorrection will consist of 4 member functions. We will see the detailed process and the relevance of those functions in this section.

Before going to the process we will look at the libraries that is used in this Class

* OpenCv : OpenCv is a popular library to do computer vision tasks in an efficient way.
* Numpy : Numpy is used to deal with multi dimensional arrays.Since the images are multi-dimensional representations of the pixel intensities.
* Math : math library allows us to use mathematical operations directly without explicitly writing them.
* Pillow : It's also a computer vision library with more features with less code.

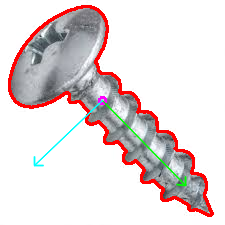
### **drawAxis**

* This function takes 5 inputs(image file, 1st set of points where you want to draw axis, 2nd set of points where you want to draw axis, color in which you want to draw axis, thickness of the line you want to draw) and will not return anything.
* This function is written to visualize the eigenvectors that are calculated in getOrentation function.



### **getOrientaion**

* This function takes 2 parameters (pts: points in which we need to find orientation, preprocessed image file) and returns the angle of the object.
* This function is written to capture the orientation of the segmented object in the image.
* It will first find the center of the region of interest(object) and it will find the eigenvectors of those points through PCA and calculate the angle between the two eigenvectors.
* The sample output of the getOreientattion function will be:



### **measure\_angle**

* This function takes the input as the path of the image and returns the angle of a particular object,in which we need to rotate.
* This function was written to preprocess the image and segment the different objects and find the angle of that object by passing that segmented image to getOrientation function.
* Note: This function was written because sometimes we may need to find the orientation of different objects in the same image so this function will help you to get the orientation of each object in the same image.

### **correct\_rotation**

* This function takes 2 arguments (The path of the input image,The path of the rotated image) and saves the output image in the given output path.
* This function uses measure\_angle function to get the angle how much to rotate and saves it in the output path given.

