# Image Processing for Gear Measurement by using Python Programming

**Akshay Sardar Patil[1], Prof. V. P. Gaikwad[2]**

*[1]M.Tech (PDD) Student, Dept. of Mech. Engineering, DKTE Society's Textile and Engg. Institute, Ichalkaranji, Maharashtra, India*
*[2]Professor, Dept. of Mech. Engineering, DKTE Society's Textile and Engg. Institute, Ichalkaranji, Maharashtra, India*

---***---

**Abstract -** *The accurate measurement of the gears plays vital role in inspection of the gears. In current scenario the tools used for inspection are either expensive or time consuming. Apart from this, some methods of measurement cannot be employed for measurements of gear. The objective in context of problem statement of this project is to develop the Image Processing System for gear profile measurement. The purpose of this project is to use computer vision technology to develop a non-contact and precise system that allows measurement of gear parameters. The python language is used to generate output results in quick time with high accuracy. The different sample gears were tested by this Image Processing system and subsequent results were achieved by python programming. The output results shows that the all gears are inspected significantly in lesser period of time. It also increases the quality and productivity of gear production with reduction in the labor cost and lead time to promote the growth of industry.*

***Key Words***: **gear parameters, image processing, python programming.**

## 1. INTRODUCTION

### 1.1. Brief Information of Image Processing System

The Image Processing System is developed as per the requirement of the concerned industry. It includes mechanical and electronic components such as conveyor system, web camera, controller, DC motor, sensor, relay module and servo motor.

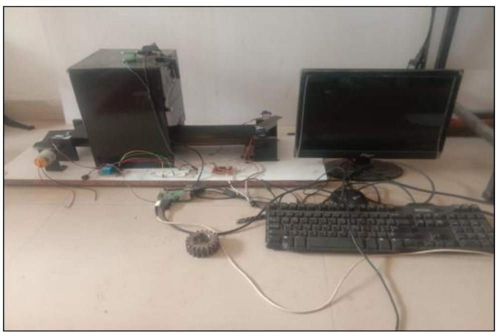

**Fig -1**: General Layout

Additionally, the Python Language is used to generate a program for measurement of gear parameters.

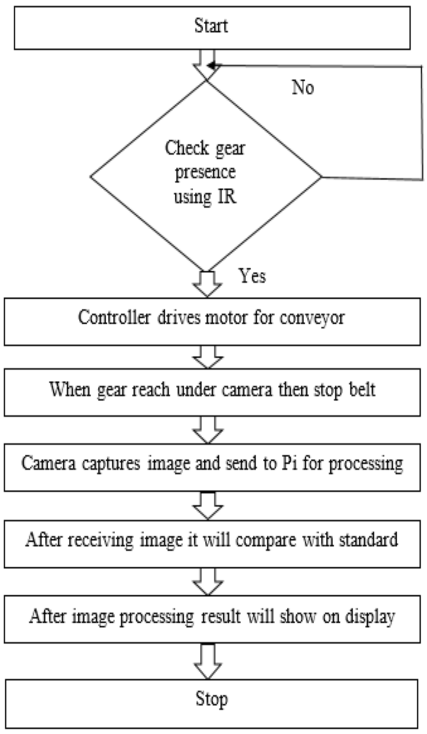


**Fig-3.** Flow chart of working of system

The work flow of Image Processing System is as shown above in the flow chart.

### 1.2. Advantages of Image Processing System

1) The measurement accuracy obtained by using image processing is high.
2) Time required for calibration of gear is less than the manual inspection.
3) It reduces labor cost because single operator can handle this system.
4) Semi -skilled person also can use this system.
5) Avoid human errors in measurement.
6) Quality assured gears can be obtained by using this system.
7) Avoid mental and physical fatigue of labor.

## 2. PROBLEM STATEMENT

After studying the available literature and methods employed for quality assurance of gear manufacturing and discussion with concerned industry the problem statement for current work is to employ image processing method with the design and development of quality assurance mechanism for gear using image to reduce manual labor and improve quality of product of concerned industry

## 3. CAD MODELING OF SYSTEM

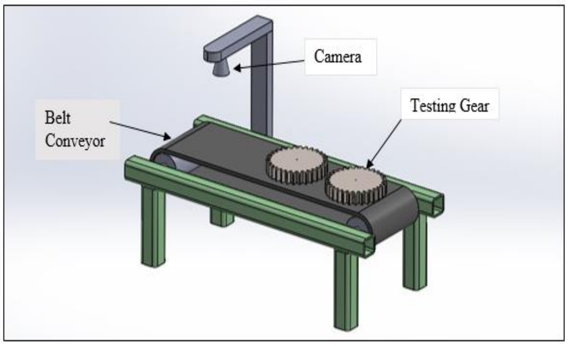


**Fig-2:** Isometric view

## 4. PYTHON PROGRAMMING

### 4.1. Introduction to Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

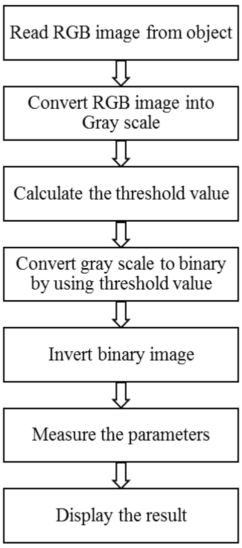


**Fig-4.** Flow chart of process of system

### 4.2. Algorithm of Program

| | |
|---|---|
| 1. Organize imports/ libraries | Input |
| 2. GIPO pin numbering | |
| 3. Input from camera | |
| 4. Apply multiple techniques | Process |
| 5. Apply Hough transformation | |
| 6. Compute weighted average | |
| 7. Counter detection | |
| 8. Thresholding of image | |
| 9. Find counter in gear edge slice | |
| 10. Calculate Fourier transformation | |
| 11. Clone the frame | |
| 12. Add padding and crop the image | |
| 13. Display the frame | Output |
| 14. Free up memory and output | |

### 4.3. Python Program

```
# organize imports/ Libraries
import cv2
import imutils
import numpy as np
import math
import scipy.fftpack
import time
import tflearn
import tensorflow as tf
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression
import RPi.GPIO as GPIO
# GPIO Pin Numbering
GPIO.setmode(GPIO.BCM)
Motor1A = 17
```

```
Motor1B = 27
sensor = 3
GPIO.setup(Motor1A,GPIO.OUT)  # All pins as Outputs
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(sensor,GPIO.IN)

# Getting input from camera
LR = 1e-3
Img_Size =50
tf.reset_default_graph()
2831pprox. = input_data(shape =[None, Img_Size, Img_Size,

1], name ='input')
2831pprox. = conv_2d(2831pprox., 32, 5, activation ='relu')
2831pprox. = max_pool_2d(2831pprox., 5)
2831pprox. = conv_2d(2831pprox., 64, 5, activation ='relu')
2831pprox. = max_pool_2d(2831pprox., 5)
2831pprox. = conv_2d(2831pprox., 128, 5, activation ='relu')
2831pprox. = max_pool_2d(2831pprox., 5)
2831pprox. = fully_connected(2831pprox., 1024, activation ='relu')
2831pprox. = dropout(2831pprox., 0.5)
2831pprox. = fully_connected(2831pprox., 2, activation ='softmax')
2831pprox. = regression(2831pprox., optimizer ='adam', learning_rate = LR,

    loss ='categorical_crossentropy', name ='targets')
model = tflearn.DNN(2831pprox., tensorboard_dir ='log')
model.load('slot.model')
str_label='new'

#Applying Multiple Techniques
def slot_detection(im):
  thresholded1=cv2.imread(im,cv2.Imread_Grayscale)
  #cv2.imshow("Thesholded", thresholded1)
  newimg = cv2.resize(thresholded1,(int(Img_Size),int(Img_Size)))
  data = newimg.reshape(Img_Size, Img_Size, 1)
model_out = model.predict([ data])[0]
model_out=list(model_out)
val=max(model_out)
idx=model_out.index(max(model_out))

print(idx)
if(idx==0):
  print("Slot present")
else:
  print("No Slot present")

#Apply Hough transform on the blurred image.
_,contours, _ = cv2.findContours(image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
contour_list = []
k=0
cropped=0
cc=0
for c in contours:
  x,y,w,h = cv2.boundingRect©
  #print(x,y,w,h)

 # Threshold the image
  #_, thresh = cv2.threshold(cv2.cvtColor(gear_slice, cv2.Color_Bgr2gray), 0 , 255, 0)

 # Find the contours in the edge_slice
  _, edge_slice_contours, _ = cv2.findContours(gear_slice, cv2.retr_external, cv2.chain_approx_simple)

# Clean up
cv2.destroyAllWindows()
print('max_dis',max(edge_dis))
print('min_dis',min(edge_dis))
max_edge=max(edge_dis)
min_edge=min(edge_dis)

# Calculate the Fourier transform
yf = scipy.fftpack.fft(distances)
# draw the segmented region and display the frame
```

```
#print("Height of teeth in mm",x[1]*0.5*0.2645)
print("outer diameter in pixel",x[2])
print("outer diameter in mm",x[2]*0.2645)
print("Inner diameter in pixel",x[3])
print("Inner diameter in mm",x[3]*0.2645)
# free up memory
camera.release()
```

## 5. RESULTS

### Sample 1:

Sample 1 shows 19 number of teeth gear in which fig. 9.1.1 shows captured raw image which read the image from input file. With the help of raw image tracing of edge takes place is shown in fig. 9.1.2

Then it is converted into binary image for thresholding and inverting the image which shown in fig. 9.1.3 and fig. 9.1.4. With reference of inverted image cropped image takes place for slot is shown in fig. 9.1.5 and fig. 9.1.6 shows output result of sample.

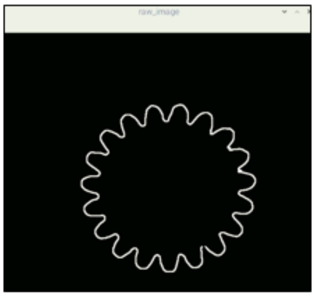

**Fig. 5. Raw**

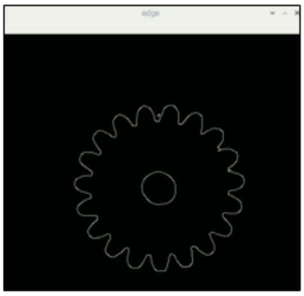

**Fig. 6. Edge**

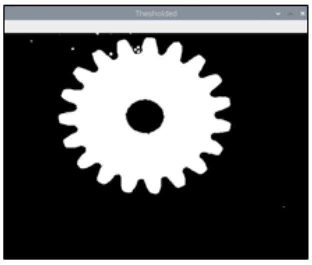

**Fig. 7. Threshold**

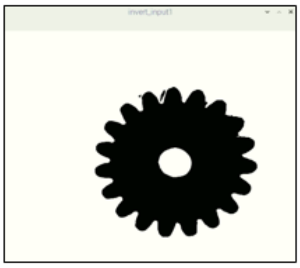

**Fig. 8. Invert input**

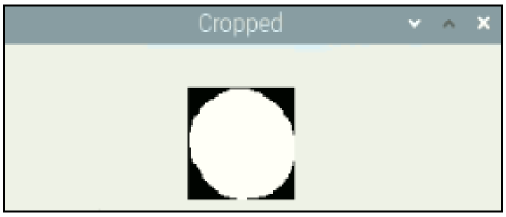


**Fig. 9. Cropped**

**Result of Python Program:**

```
*Python 3.7.3 Shell*
File Edit Shell Debug Options Window Help
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tflearn/objective
s.py:66: calling reduce_sum_v1 (from tensorflow.python.ops.math_ops) with keep_d
ims is deprecated and will be removed in a future version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tflearn/summaries
.py:46: The name tf.summary.scalar is deprecated. Please use tf.compat.v1.summar
y.scalar instead.

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow_core/p
ython/ops/math_grad.py:1251: add_dispatch_support.<locals>.wrapper (from tensorf
low.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tflearn/helpers/t
rainer.py:134: The name tf.train.Saver is deprecated. Please use tf.compat.v1.tr
ain.Saver instead.

WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow_core/p
ython/training/saver.py:1276: checkpoint_exists (from tensorflow.python.training
.checkpoint_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
Press any key to calibrate background
ready to detect put gear
max_dis 168.23792675850473
min_dis 130.86252328302402
Number of teeth 19
Height of teeth in pixel 18.687701737740355
Height of teeth in mm 4.942897109632324
outer diameter in pixel 168.23792675850473
outer diameter in mm 44.4989316276245
Inner diameter in pixel 130.86252328302402
Inner diameter in mm 34.613137408359854
Number of teeth is ok
Height of teeth is ok
outer diameter is ok
Inner diameter is ok
1
No slot present
```

**Figure 10.** Output of Program

**Table 1. Results of Sample gears in Tabular Form**

| Sr. no. | Company's data | | | | Actual reading by image processing | | | | Remark |
|---|---|---|---|---|---|---|---|---|---|
| | Teeth | Tooth Height (mm) | OD (mm) | ID (mm) | Teeth | Tooth Height (mm) | OD (mm) | ID (mm) | |
| 1. | 19 | 5 | 45 | 35 | 19 | 4.94 | 44.98 | 34.61 | Accepted |
| 2. | 21 | 4 | 35 | 28 | 21 | 3.55 | 34.65 | 27.54 | Accepted |
| 3. | 23 | 5 | 50 | 40 | 23 | 4.97 | 49.98 | 40.04 | Accepted |
| 4. | 24 | 4 | 46 | 38 | 24 | 4.14 | 45.75 | 37.46 | Accepted |
| 5. | 36 | 4 | 58 | 50 | 36 | 3.90 | 58 | 50.19 | Accepted |
| 6. | 19 | 5 | 45 | 35 | 17 | 4.84 | 44.69 | 34.40 | Rejected |
| 7. | 23 | 5 | 50 | 40 | 21 | 5.17 | 50.16 | 39.61 | Rejected |

## 6. CONCLUSIONS

The goal of the research is to reduce the gear parameter measurement time and remove the physical exhaustion of labor problem in industries. It was done by using Image processing which not only display the result but also sorting of gear also done. The conclusions from this dissertation work are given as follows,

1. As compare to manual, image processing shows faster results i.e. shows results in half time than the manual.
2. More than one gear can placed on belt so, gear handling time reduced.
3. Inspection rate of image processing is almost double than the manual method.
4. Image processing has resulted in reduced errors in gear sorting. The defective gears are accurately removed.
5. The experimental designed system is simple, stable and reliable with good speed and high accuracy.

## 7. FUTURE SCOPE

1. If we use high quality camera i.e. high pixel resolution, it will help to meet more accurate values/readings.
2. More stable light intensity source improve calibration of background.
3. We can control whole system by IoT with some modifications.
4. Other parameters of gear may be possible with this method by some additional sensors like laser sensor, tilt sensor.

## REFERENCES

[1] Manting Luo, Shuncong Zhong (2018) “Non- contact measurement of small module gear using optical coherence tomology” Journal of Applied Sciences for mechanical Engineering, China, December 4, 2018.

[2] M. N. Dhavalikar, Ankit Bhat (2018) “Geometric verification of gears using image processing technique” Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-4, Page no. 1644-1654.

[3] Ruiling Lui, Dexing Zhong (2016) “A bevel gear quality inspection system based on camera vision technology” Journal of multidisciplinary digital publishing institute, China, Volume-16, August 2016, Page No. 1-17.

[4] Wencheng Wang, Jian Li (2015) “Measurement system of gear parameters based on machine vision” Journal of Measurement and Control, Weifang, P.R. China, Volume-48, October 2015, Page no. 242-248.

[5] Wencheng Wang, Xiaojun Cui (2014); “An image-based system for measuring workpieces” Journal of Measurement and Control, Volume-47, November 2014, Page no. 283-287.

[6] Haque Nawaz, Himat Ali (2014) “Gear measurement using image processing in matlab” Journal of Innovative Technology, University of Sciences and Technology, Karachi, Volume-3, January 2014.

[7] Amandeep Mavi, Mandeep Kaur (2012) “Identify defects in gears using digital image processing” International Journal of Engineering Research and Development, Volume-1, June 2012, Page no. 49-55.

[8] Wang Wen-cheng and Ji Tao (2011) “A system design method for gear parameters analyzing” Future Intelligent Information Systems, (2011), Page no. 7-13.

[9] M.A.Younes (2005) “Automatic measurement of spur gear dimensions using laser light” Journal of Optical Engineering, Alexandria University, Egypt, Volume-44, October 2005.

[10] M J Robinson, J P Oakley and M J Cunningham (1995) “The accuracy of image analysis methods in spur gear metrology” Measurement Science Technology, UK, March 1995, Page no. 860-871.

[11] https://www.geeksforgeeks.org

[12] https://stackabuse.com

[13] https://www.raspberrypi.org

[14] https://elprocus.com

[15] https://gearseds.com