Optical Character Recognition using OpenCV

Sathish Kasilingam

Issues to be tackled:

- Text at angles
- GD&T Symbols
- Auto recognition of text areas in images
- Higher accuracy

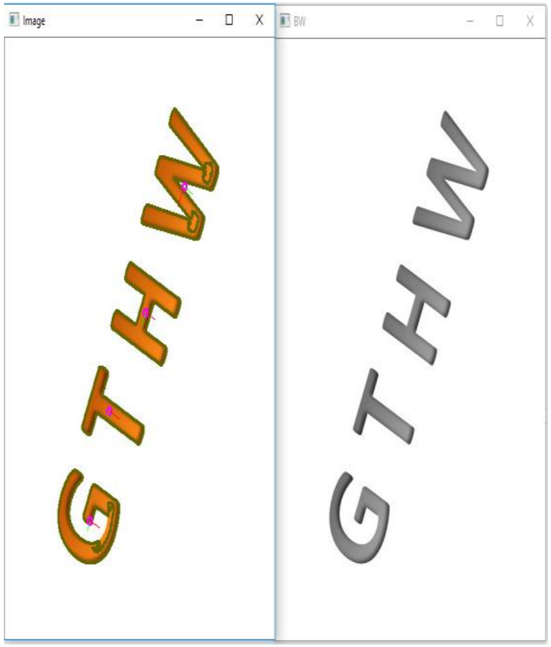
Text at angles-Solution:

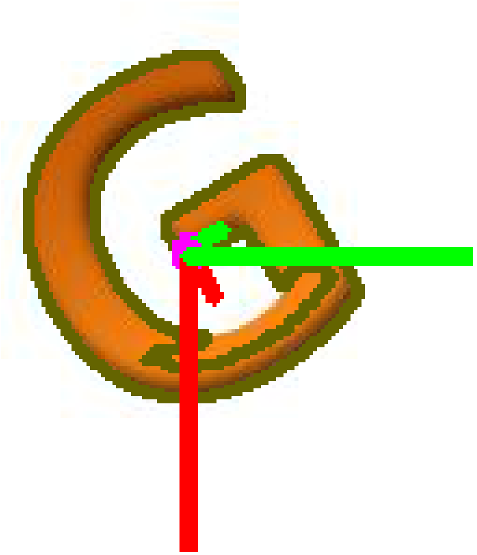
- Determining the angles by using principal component analysis
  - The contours in the image are determined
  - Principal component analysis yields the directions of highest data distribution and lowest data distribution
  - The vector characterizing the direction of highest data distribution is selected
  - The angle the vector subtends with the horizontal is determined
  - This angle is used to determine the rotation matrix for that character set

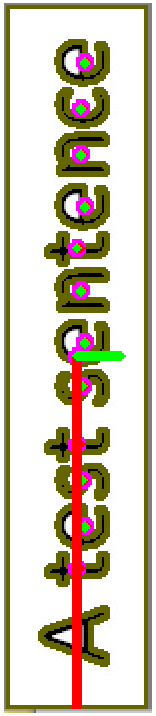
Output: A horizontal image to be used in OCR

In Progress

Text at angles-Solution:







Green Vector: Vector characterizing the direction of highest data distribution

Red Vector: Vector characterizing the direction of lowest data distribution

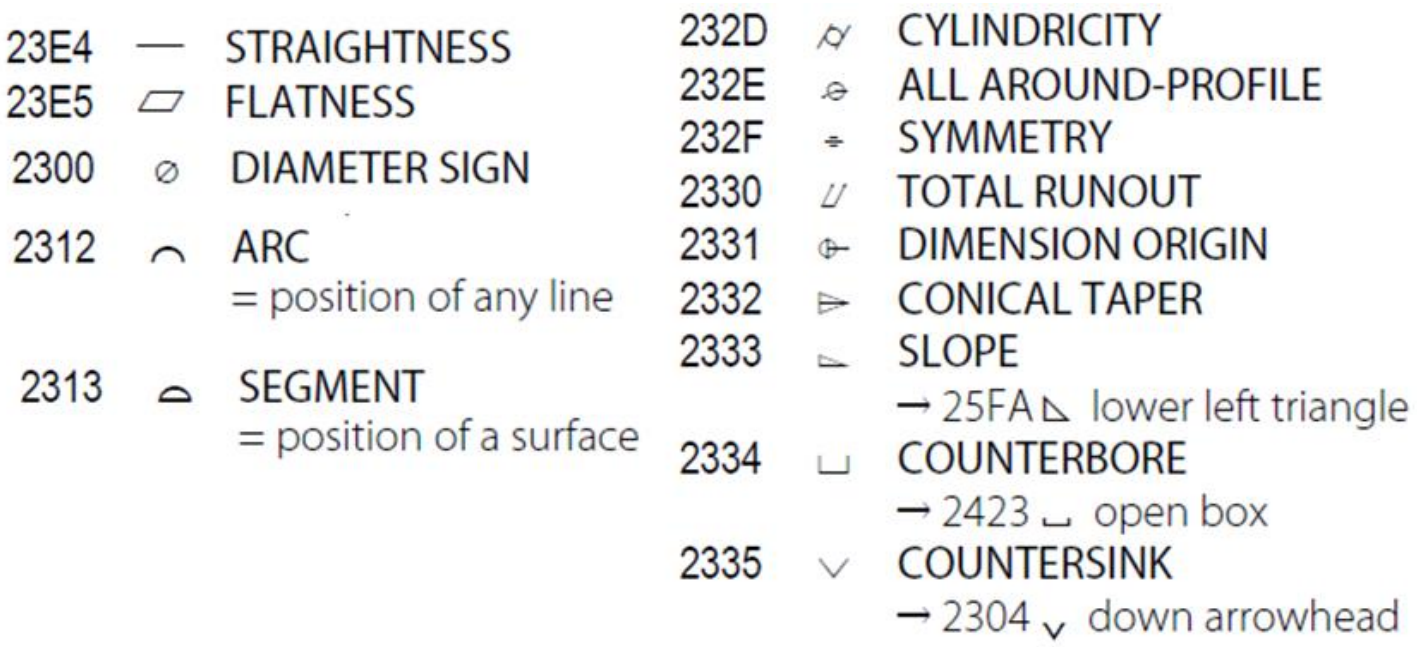
GD&T Symbols-Solution:

- Recognizing and Printing GD&T symbols
- A set is used to train the data. The needed characters can be entered in by entering their ASCII Code or Unicode.

Output: The XML files that are to be used an inputs for OCR

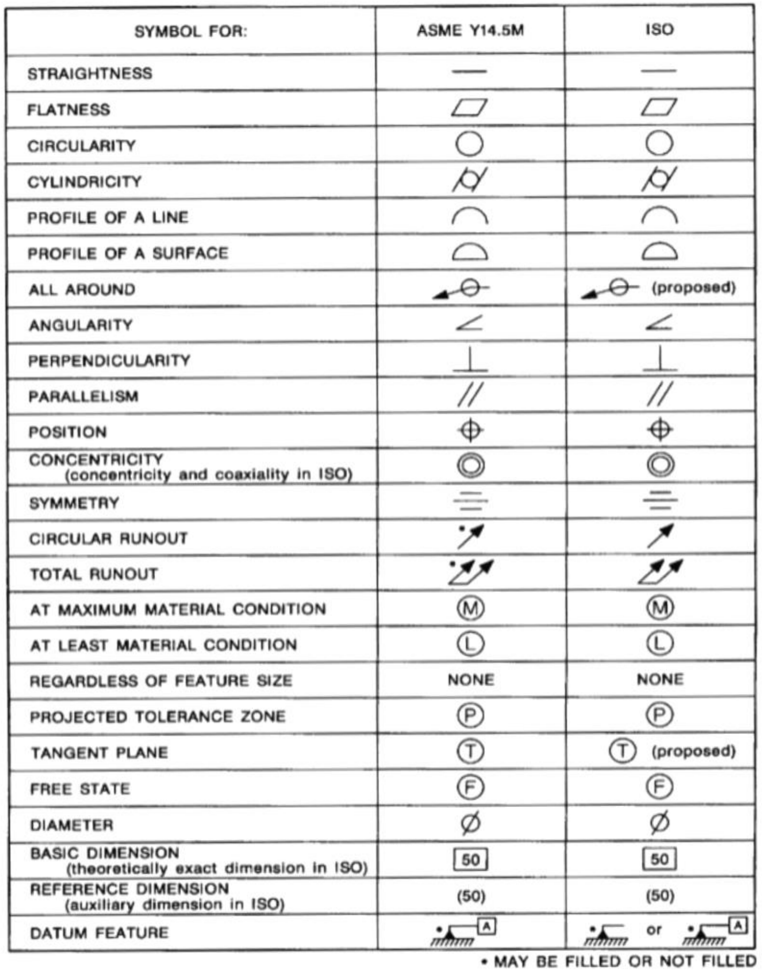
In Progress for all symbols

GD&T Symbols-Solution:



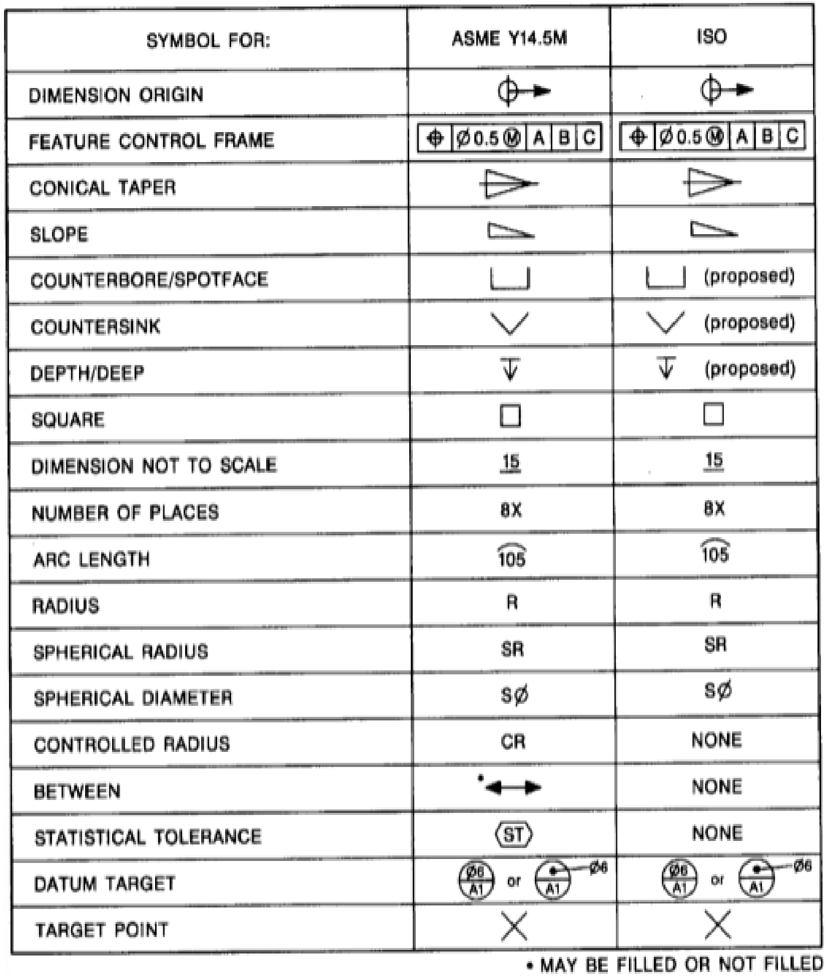
23E4	⏤	STRAIGHTNESS
23E5	⏥	FLATNESS
2300	⌀	DIAMETER SIGN
2312	⌒	ARC = position of any line
2313	⌓	SEGMENT = position of a surface
232D	⌭	CYLINDRICITY
232E	⌮	ALL AROUND-PROFILE
232F	⌯	SYMMETRY
2330	⌰	TOTAL RUNOUT
2331	⌱	DIMENSION ORIGIN
2332	⌲	CONICAL TAPER
2333	⌳	SLOPE → 25FA ◺ lower left triangle
2334	⌴	COUNTERBORE → 2423 ␣ open box
2335	⌵	COUNTERSINK → 2304 ⌄ down arrowhead

GD&T Symbols:



SYMBOL FOR:	ASME Y14.5M	ISO
STRAIGHTNESS	⏤	⏤
FLATNESS	⏥	⏥
CIRCULARITY	○	○
CYLINDRICITY	⌭	⌭
PROFILE OF A LINE	⌒	⌒
PROFILE OF A SURFACE	⌓	⌓
ALL AROUND	←⊖	←⊖ (proposed)
ANGULARITY	∠	∠
PERPENDICULARITY	⊥	⊥
PARALLELISM	//	//
POSITION	⌖	⌖
CONCENTRICITY (concentricity and coaxiality in ISO)	◎	◎
SYMMETRY	⌯	⌯
CIRCULAR RUNOUT	•↗	↗
TOTAL RUNOUT	•⌰	⌰
AT MAXIMUM MATERIAL CONDITION	Ⓜ	Ⓜ
AT LEAST MATERIAL CONDITION	Ⓛ	Ⓛ
REGARDLESS OF FEATURE SIZE	NONE	NONE
PROJECTED TOLERANCE ZONE	Ⓟ	Ⓟ
TANGENT PLANE	Ⓣ	Ⓣ (proposed)
FREE STATE	Ⓕ	Ⓕ
DIAMETER	Ø	Ø
BASIC DIMENSION (theoretically exact dimension in ISO)	$\boxed{50}$	$\boxed{50}$
REFERENCE DIMENSION (auxiliary dimension in ISO)	(50)	(50)
DATUM FEATURE	•▲ A	•▲ or •▲ A

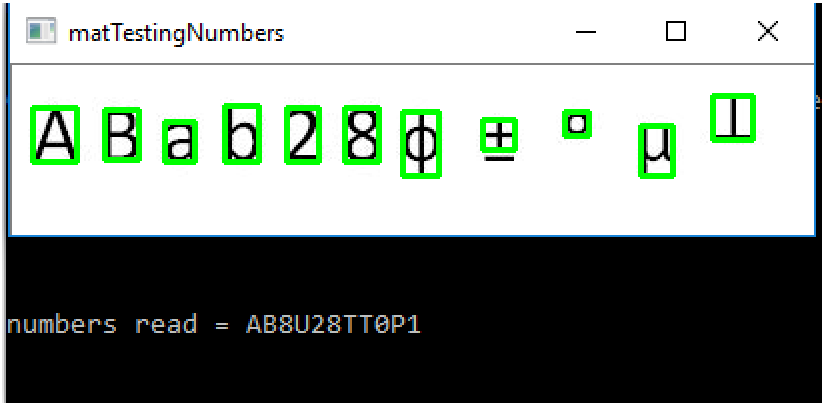
• MAY BE FILLED OR NOT FILLED



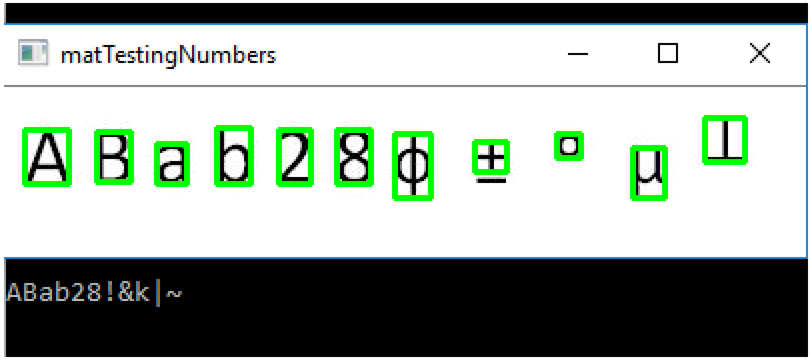
SYMBOL FOR:	ASME Y14.5M	ISO
DIMENSION ORIGIN	⌾→	⌾→
FEATURE CONTROL FRAME	⌖ | Ø0.5 Ⓜ | A | B | C	⌖ | Ø0.5 Ⓜ | A | B | C
CONICAL TAPER	⌲	⌲
SLOPE	⌳	⌳
COUNTERBORE/SPOTFACE	⌴	⌴ (proposed)
COUNTERSINK	⌵	⌵ (proposed)
DEPTH/DEEP	↧	↧ (proposed)
SQUARE	□	□
DIMENSION NOT TO SCALE	$\underline{15}$	$\underline{15}$
NUMBER OF PLACES	8X	8X
ARC LENGTH	$\overset{\frown}{105}$	$\overset{\frown}{105}$
RADIUS	R	R
SPHERICAL RADIUS	SR	SR
SPHERICAL DIAMETER	SØ	SØ
CONTROLLED RADIUS	CR	NONE
BETWEEN	•↔	NONE
STATISTICAL TOLERANCE	⟨ST⟩	NONE
DATUM TARGET	Ø6 A1 or A1 Ø6	Ø6 A1 or A1 Ø6
TARGET POINT	×	×

• MAY BE FILLED OR NOT FILLED

GD&T Symbols:

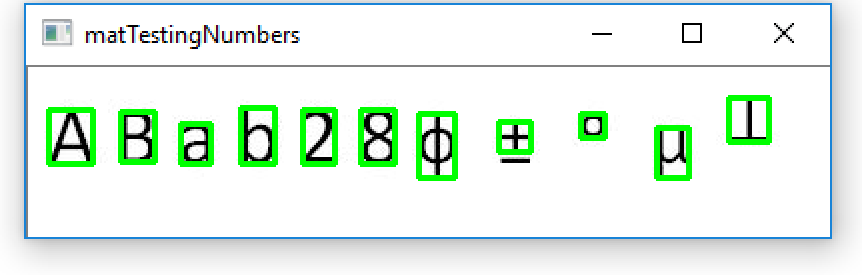


```
numbers read = AB8U28TT0P1
```



```
ABab28!&k|~
```

ABab28ϕ±kμ⊥



Auto recognition of text areas in images-Solution:

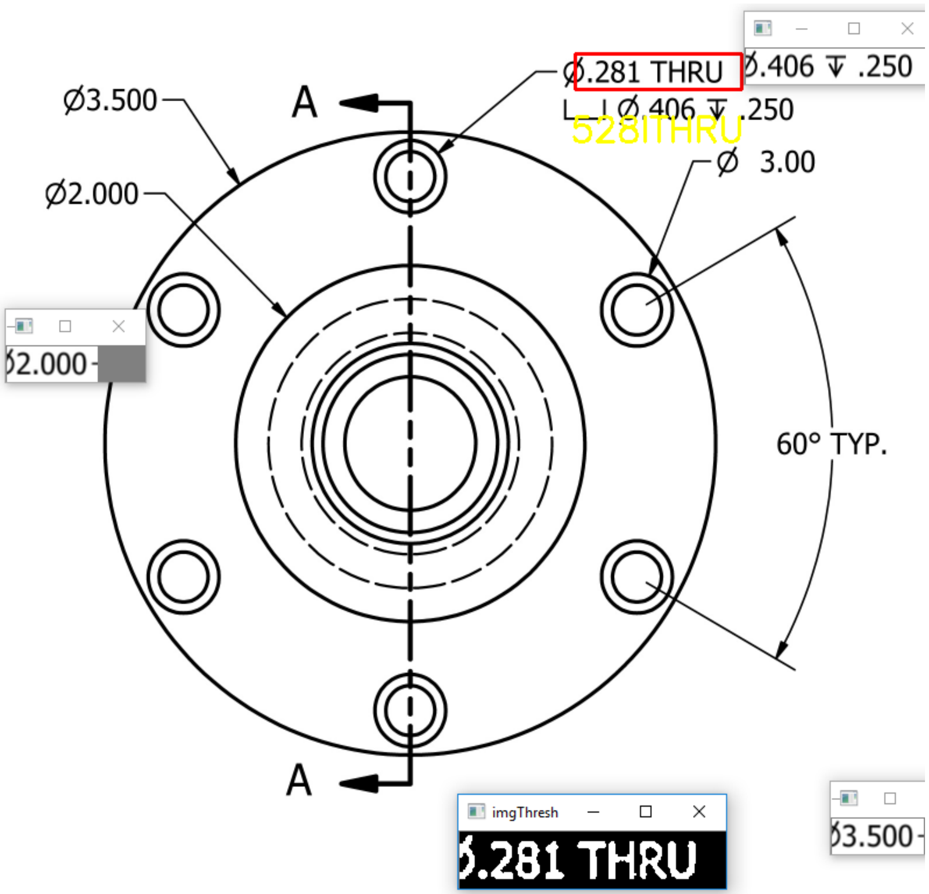
- Determining the character sets
  - The contours in the image are determined.
  - They care compared with characters
  - The sets of characters are classified out of the image
  - The sets are sorted according to length
  - The region in the image is determined and then highlighted

Output:

- The input image with the character sets highlighted.
- Individual sets that are made horizontal and then can be used in OCR.

Nearly Done

Auto recognition of text areas in images-Solution:



```
528ITHRU
```

```
Ø.281 THRU
```

```
Ø.406 ⊽ .250
```

```
Ø2.000
```

```
Ø3.500
```

Higher Accuracy-Solution:

- Loading the character sets in Unicode format
  - All the characters can be assigned a Unicode character
  - Some GD&T symbols have Unicode characters
  - Increasing the training dataset with more fonts and characters
  - Using the knn algorithm speeds up and also accurately determines characters.

Output:

- An XML file that has the Unicode references of the characters.
- An XML file that has the pixel intensity values of the characters.

In Progress-Requires all character sets

Thank You