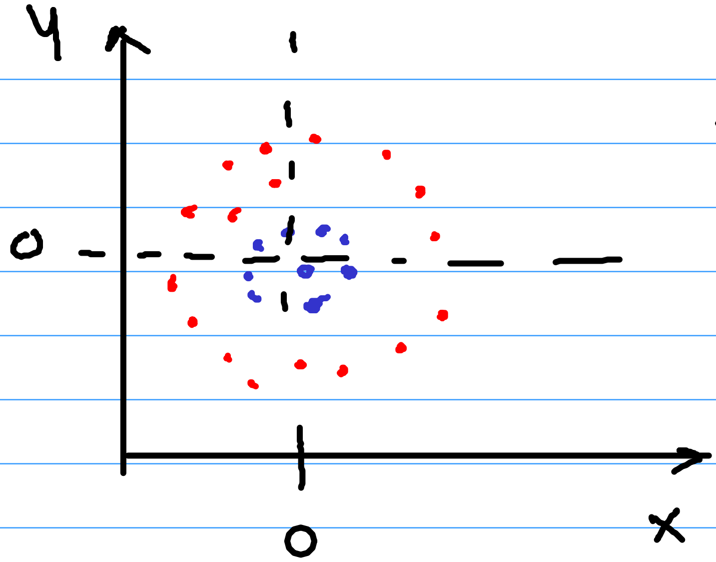
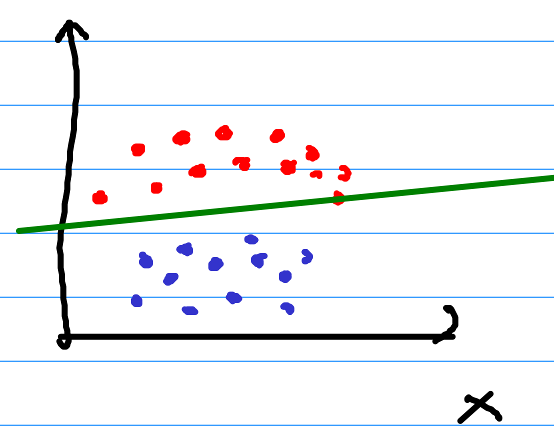
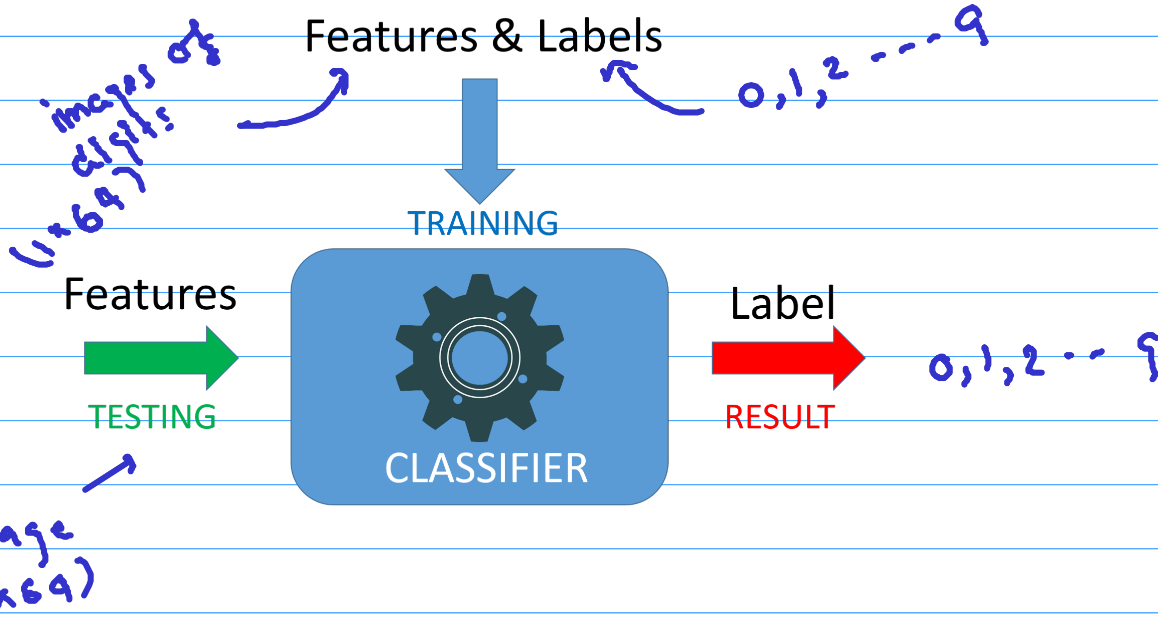
$z = x^2 + y^2$ ← polynomial kernel





SVM for Handwritten Digits



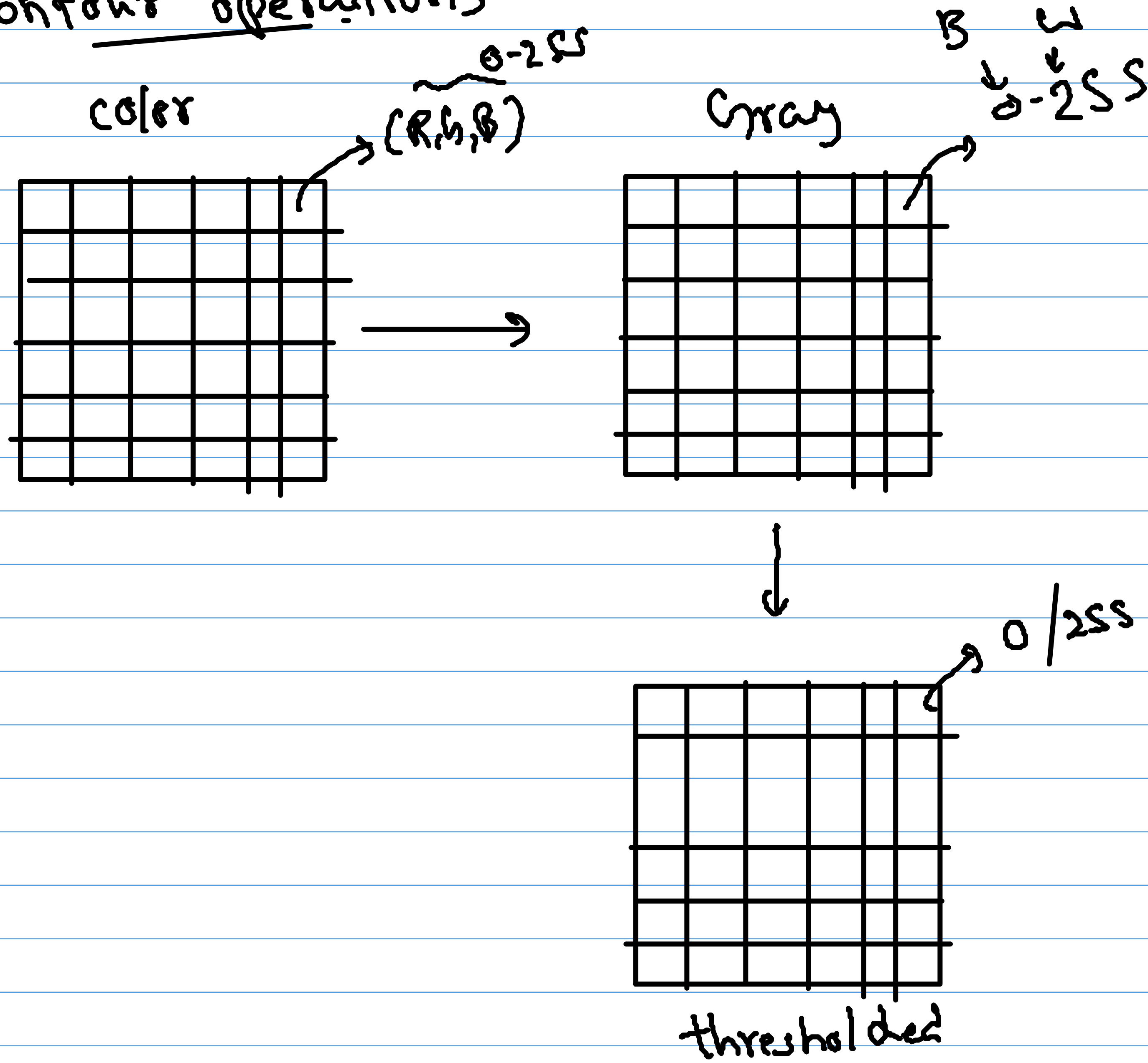
✓ pip install pillow

$0 \rightarrow 0/255 = 0 \times 15 = 0$

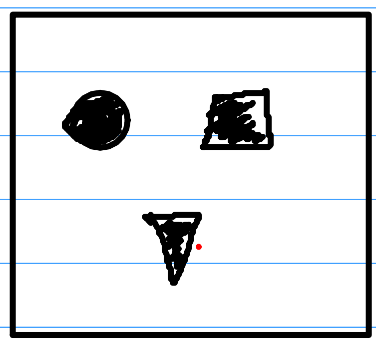
$255 \rightarrow 255/255 = 1 \times 15 = 15$

$128 \rightarrow 128/255 \approx 0.5 \times 15 = 7.5$

contour operations



contours



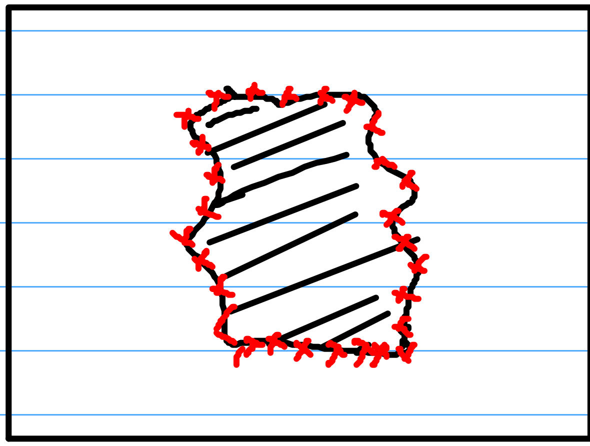
Boundary points

Contours = [[[ ],[ ],[ ]],
[[ ],[ ],[ ]],
[[ ],[ ],[ ]]]

x,y

rows → # contours

Boundry points



```
(4, 1, 2)
[[[ 21 356]]

 [[ 21 391]]

 [[ 56 391]]

 [[ 56 356]]]
```

```python
cnt=contours[0]

print(cnt.shape)
print(cnt)

for val in cnt:
    (x,y)=val[0]
    cv2.circle(img,(x,y),5,(0,255,255),-1)
```

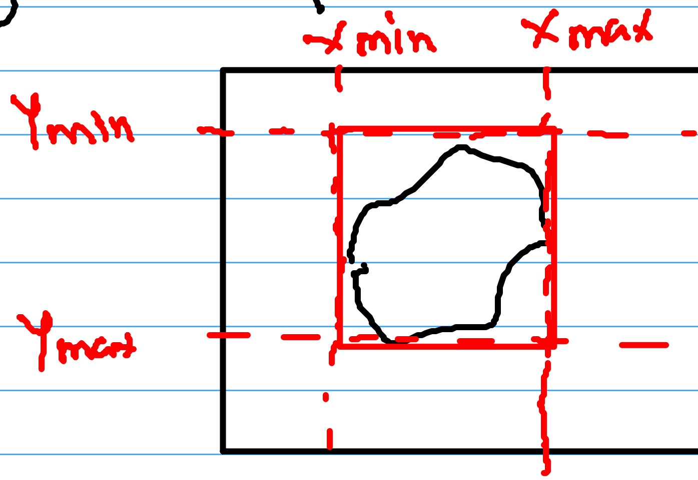
val = [[21,356]] → (1,1,2)

(x,y) = (21,356)

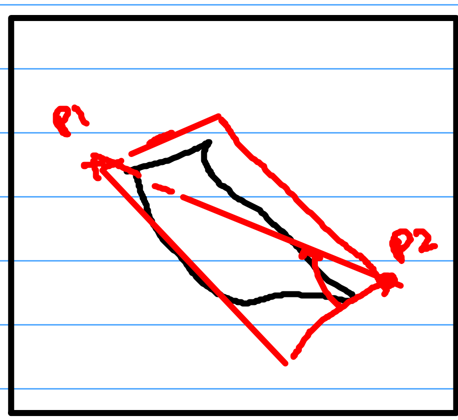
cnt=contours[0]
print(cnt.shape)
print(cnt)
for val in cnt:
(x,y)=val[0]
cv2.circle(img,(x,y),5,(0,255,255),-1)

Bounding Rectangle

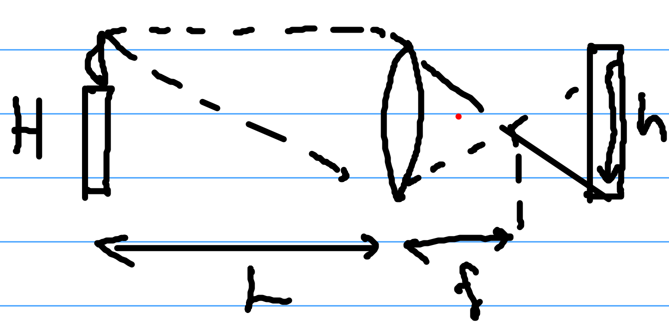
(i) Straight Rectangle



(ii) Rotated Rectangle



Finding the real area of object



$$\frac{H}{h} = \frac{L}{l}$$

ps - pixel size

The triangle similarity goes something like this: Let's say we have a marker or object with a known width *W*. We then place this marker some distance *D* from our camera. We take a picture of our object using our camera and then measure the apparent width in pixels *P*. This allows us to derive the perceived focal length *F* of our camera:

$F = (P \times D) / W$

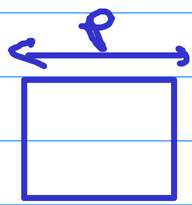
$$F = P \times D / W \rightarrow \frac{F}{D} = \frac{a}{A}$$

Focal length

photo width

Distance to object

actual width



$$A = \frac{a \cdot D}{F} \times PS$$

camera parameter

$(A) \rightarrow PS$

Stereo Vision

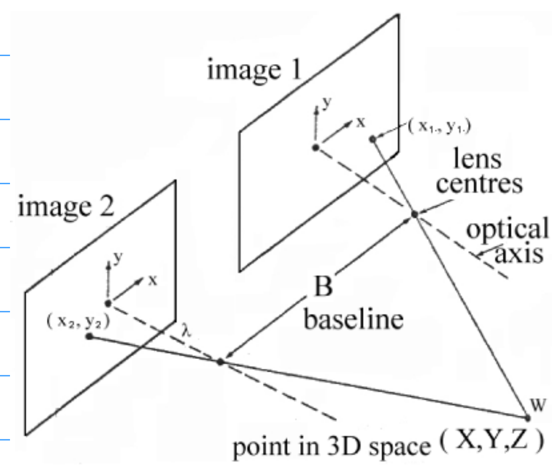
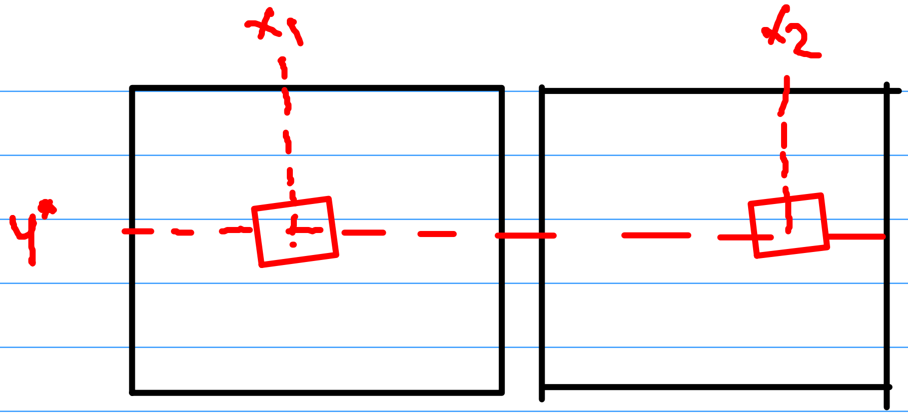


Figure 2. Stereo image geometric model.

$$\boxed{D = \frac{f * b}{d * ps}}$$

f=Focal length
b=Baseline
d=Disparity value
ps=Pixel size
D=Depth

disparity:



$d = |x_1 - x_2|$

Decision Tree Algorithms

ENTROPY = $(\frac{3}{8}) \log_2(\frac{3}{8}) + (\frac{2}{8}) \log_2(\frac{2}{8}) + (\frac{1}{8}) \log_2(\frac{1}{8}) + (\frac{2}{8}) \log_2(\frac{2}{8})$

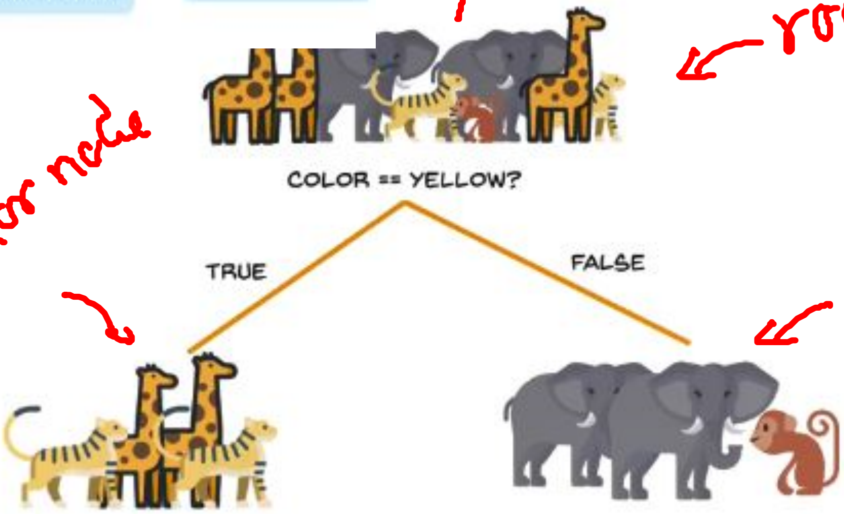
ENTROPY=0.571

$0.571 = \mathcal{E}0$

root node

interior node

interior node



①

$\mathcal{E}12$

$$\mathcal{E}11 = \sum_{i=0}^{k} p(i) \cdot \log_2(p_i) = \frac{2}{5} \cdot \log_2\left(\frac{2}{5}\right) + \frac{3}{5} \log_2\left(\frac{3}{5}\right)$$

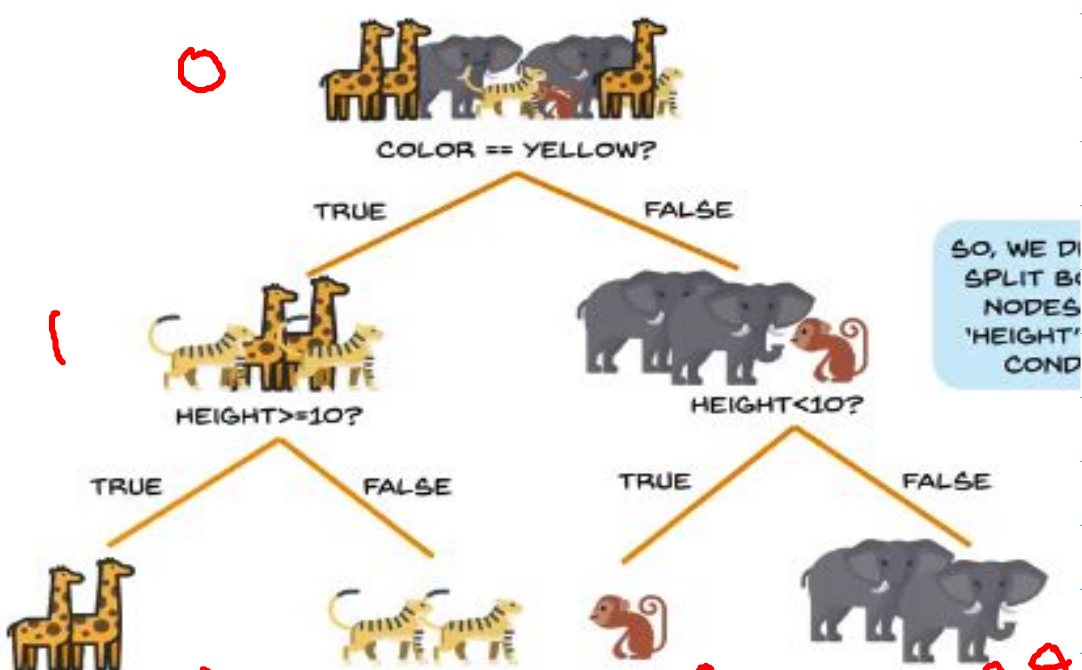
$$\mathcal{E}21 = \frac{2}{3} \log_2(2/3) + 1/3 \log_2(1/3)$$

$$\mathcal{E}1 = \mathcal{E}11 + \mathcal{E}21 = -1.889$$

0

1

2



$\mathcal{E}21$ $\mathcal{E}22$ $\mathcal{E}23$ $\mathcal{E}24$

$E2 = E21 + E23 + E22 + E24$

$$H(x) = -\sum_{for\ k \in target} (P(x = k) * log_2(P(x = k)))$$

$E21 = \frac{2}{2} \log_2 (2/2) = 0$

$E22 = 2/2 \log_2 (2/2) = 0$

$E23 = 1/1 (\log_2 (1/1)) = 0$

$E2 = 0$

$E24 = 0$

Info gain

$IG1 = E0 - E1$

$IG2 = E2 - E1$

$$H(x) = -\sum_{for\ k \in target} (P(x = k) * log_2(P(x = k)))$$

Root node

E2, G3, M1, T2

$$\varepsilon_0 = \frac{2}{8}\log\left(\frac{2}{8}\right) + \frac{3}{8}\log\left(\frac{3}{8}\right) + \frac{1}{8}\log\left(\frac{1}{8}\right) + \frac{2}{8}\log\left(\frac{2}{8}\right)$$

$\varepsilon_0 = 1.91$

(1)' condition, (color == Yellow)

G3, T2

$$\varepsilon_{11} = \frac{3}{5}\log\left(\frac{3}{5}\right) + \frac{2}{5}\log\left(\frac{2}{5}\right) = 0.97$$

E2, M1

$$\varepsilon_{12} = \frac{2}{3}\log\left(\frac{2}{3}\right) + \frac{1}{3}\log\left(\frac{1}{3}\right) = 0.92$$

$\varepsilon_1 = 1.89$

$IG_0 = \varepsilon_0 - \varepsilon_1 = 0.02$

(1)'' condition (color == gray)

E2

G3, M1, T2

$\varepsilon_{11} = 0$

$$\varepsilon_{12} = \frac{3}{6}\log\left(\frac{3}{6}\right) + \frac{1}{6}\log\left(\frac{1}{6}\right) + \frac{2}{6}\log\left(\frac{2}{6}\right)$$

$\varepsilon_{12} = 1.46$

$IG_0' = 1.91 - 1.46 = 0.35$

* * Objective → to maximize info/gain

$\varepsilon_0 = ?$

S - 50, Vg - 50, Vr - 50

S/L (3.1 - 5.6)

condition $S/L \geq 3.1$

True | False

ε_{11} | ε_{12}

$S/W \geq 5.0$