The Washington Times

Lloyd George Announces

ENGLAND AND AMERICA ARE AGREED ON ALL QUESTIO

WAR MATERIAL FROM DEPOTS EAST OF RHINE SURRENDERED

ISHIDA WILL COMMIT HARI KARI IF SANE, SAYS FRIEND

War Cost the World $175,000,000,000

PRESIDENT IS AS ENGLAND PROPOS BACK U. S. PEAC

THEY'LL HAVE TO ASK FATHER

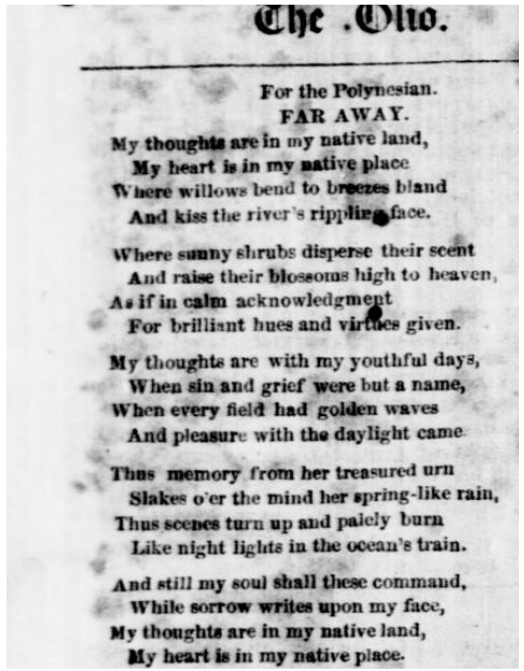
SEE PEACE DELAY IN EBERT'S DEFEAT

Premier Lloyd George Faces Tory Majority in Parliament

TODAY

3 DROWN TRYING TO RESCUE SLED

4 CROSS COUNTRY FLYERS DUE JAN. 4



The Ohio.

For the Polynesian.

FAR AWAY.

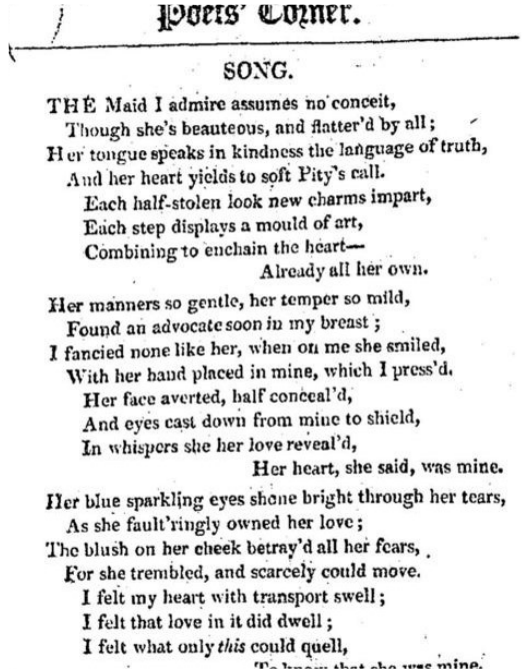
My thoughts are in my native land,
My heart is in my native place
Where willows bend to breezes bland
And kiss the river's rippling face.

Where sunny shrubs disperse their scent
And raise their blossoms high to heaven,
As if in calm acknowledgment
For brilliant hues and virtues given.

My thoughts are with my youthful days,
When sin and grief were but a name,
When every field had golden waves
And pleasure with the daylight came.

Thus memory from her treasured urn
Slakes o'er the mind her spring-like rain,
Thus scenes turn up and palely burn
Like night lights in the ocean's train.

And still my soul shall these command,
While sorrow writes upon my face,
My thoughts are in my native land,
My heart is in my native place.



Poets' Corner.

SONG.

THE Maid I admire assumes no conceit,
Though she's beauteous, and flatter'd by all;
Her tongue speaks in kindness the language of truth,
And her heart yields to soft Pity's call.
Each half-stolen look new charms impart,
Each step displays a mould of art,
Combining to enchain the heart—
Already all her own.

Her manners so gentle, her temper so mild,
Found an advocate soon in my breast;
I fancied none like her, when on me she smiled,
With her hand placed in mine, which I press'd.
Her face averted, half conceal'd,
And eyes cast down from mine to shield,
In whispers she her love reveal'd,
Her heart, she said, was mine.

Her blue sparkling eyes shone bright through her tears,
As she fault'ringly owned her love;
The blush on her cheek betray'd all her fears,
For she trembled, and scarcely could move.
I felt my heart with transport swell;
I felt that love in it did dwell;
I felt what only *this* could quell,

Image Based Classifier for Detecting Poetic Content

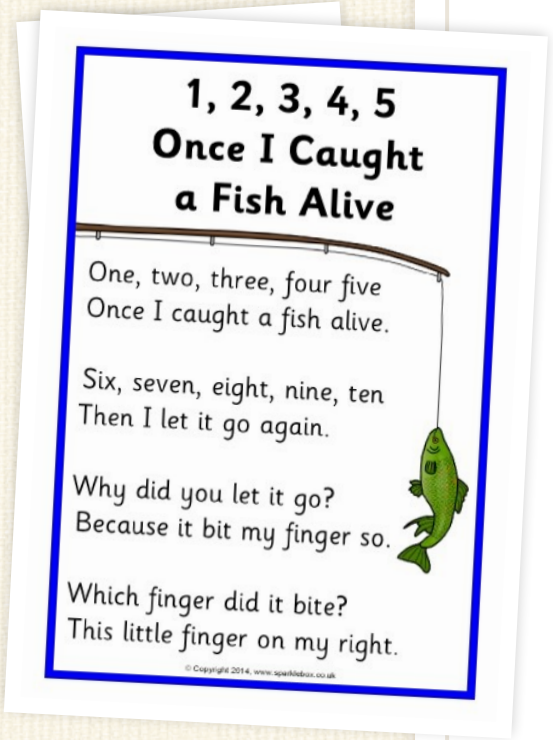
Can we find poetic content in historic newspapers based on **visual signals** alone?

Motivation

- Advance work on the use of digital images
- Making data more readily available for study

Why poetry?

- Scale
- Visual distinctness
- Interest and significance



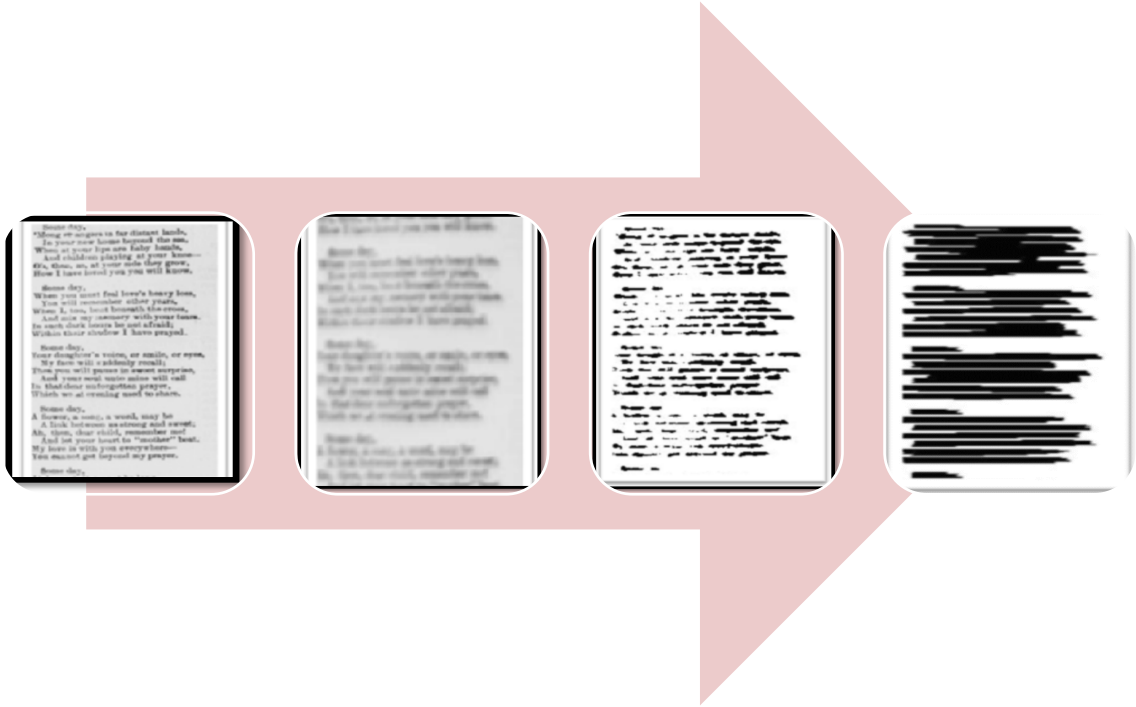
Visual features of a poem

- Whitespace between stanzas
- Content blocks with jagged right-side edges/ varying line lengths
- Left margin whitespace

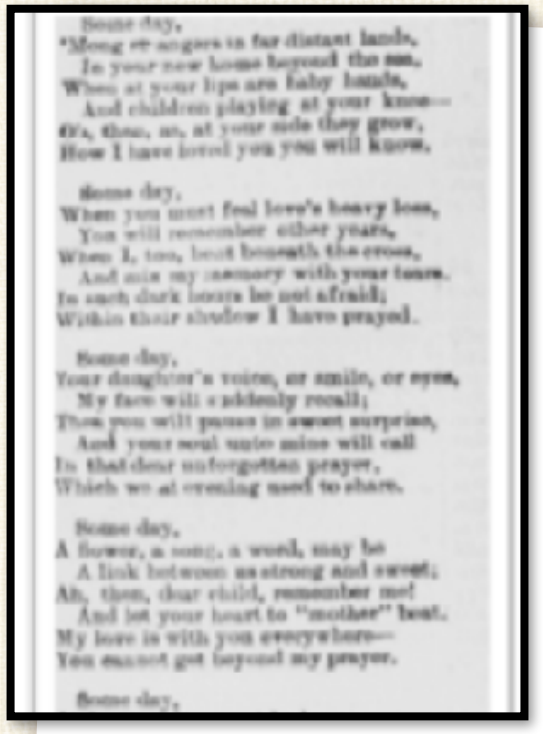
Teaching a computer to see poetry

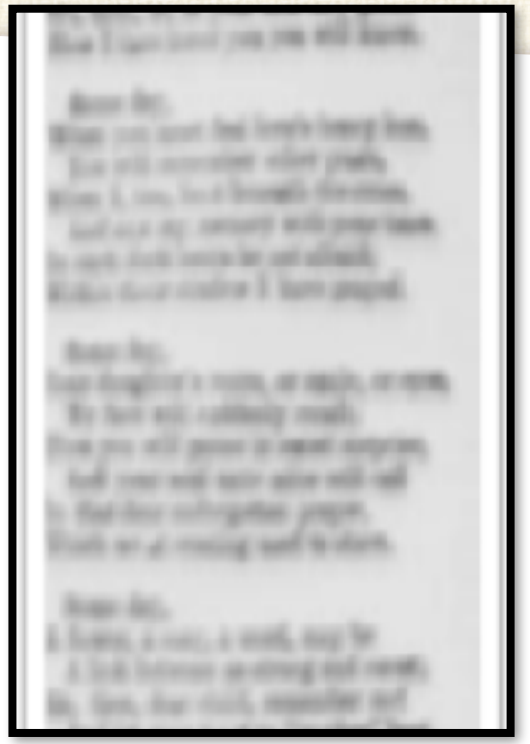
- Pre-processing
- Features extraction
- Using artificial neural network

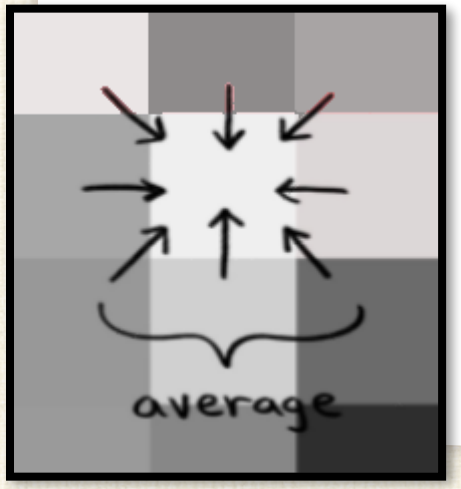
Pre-Processing Stage



- Blurring
- Bi-Gaussian binarization
- Pixel consolidation



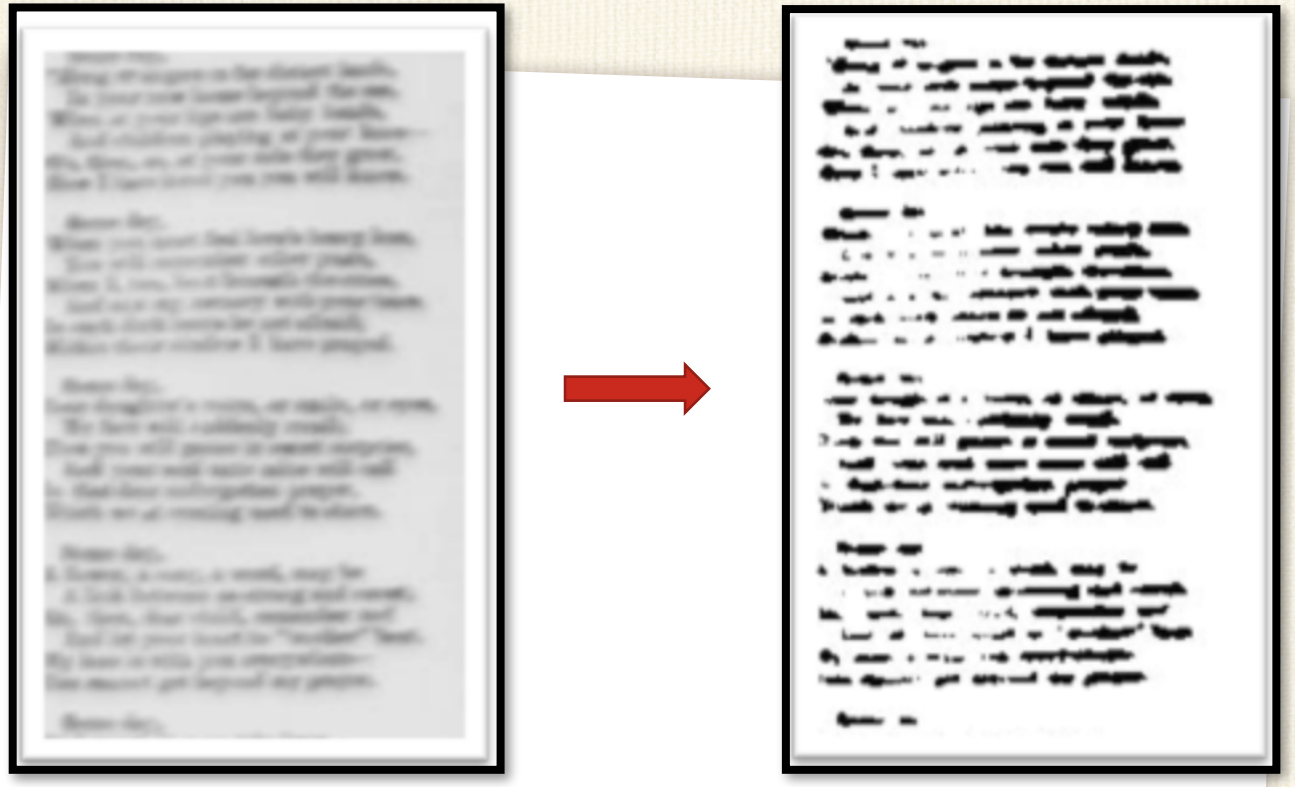




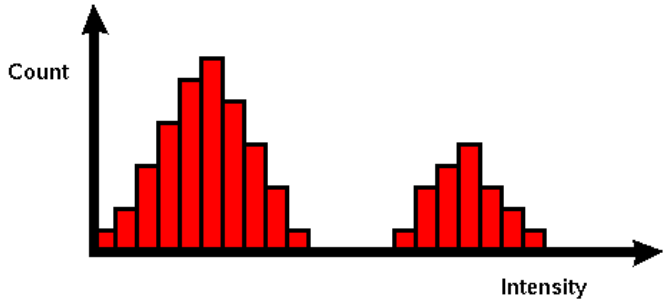
RGB

Bit	7	6	5	4	3	2	1	0
Data	R	R	R	G	G	G	B	B

Blurring



```
public void convertToBinaryImage(){
    for (int i = 0; i < this.vertical; i++){
        for(int j= 0; j < this.horizontal; j++){
            if(this.blurredImagePixels[i][j] < this.threshold){
                binaryImagePixels[i][j] = 0;
            } else {
                binaryImagePixels[i][j] = 255;
            }
        }
    }

}
```

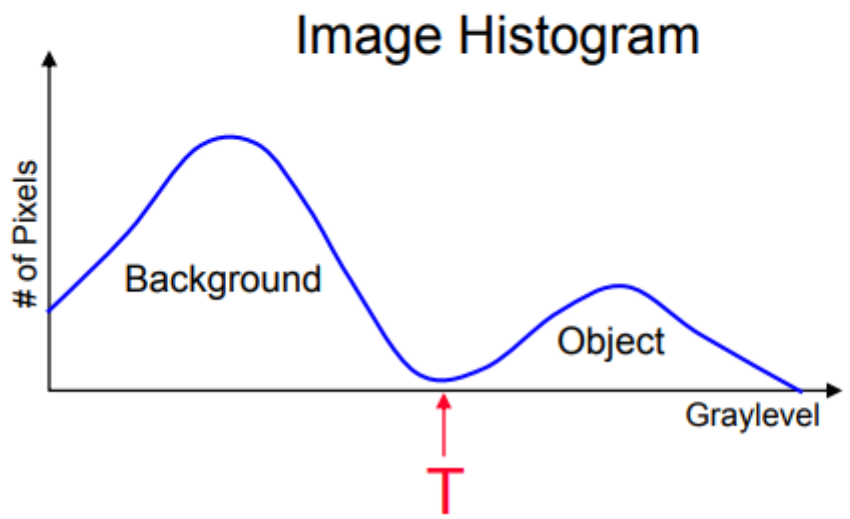


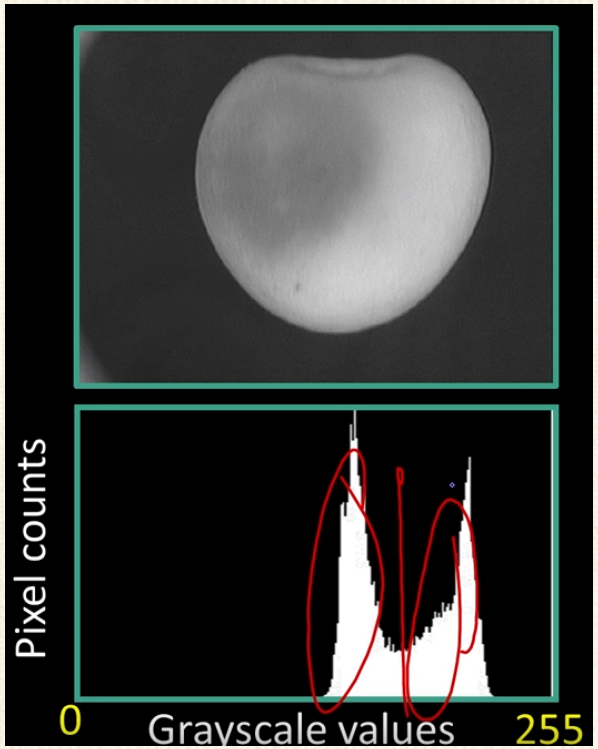
Common Names: Histogram

Binarization

Finding the threshold

Global Thresholding = Choose threshold T that separates object from background.



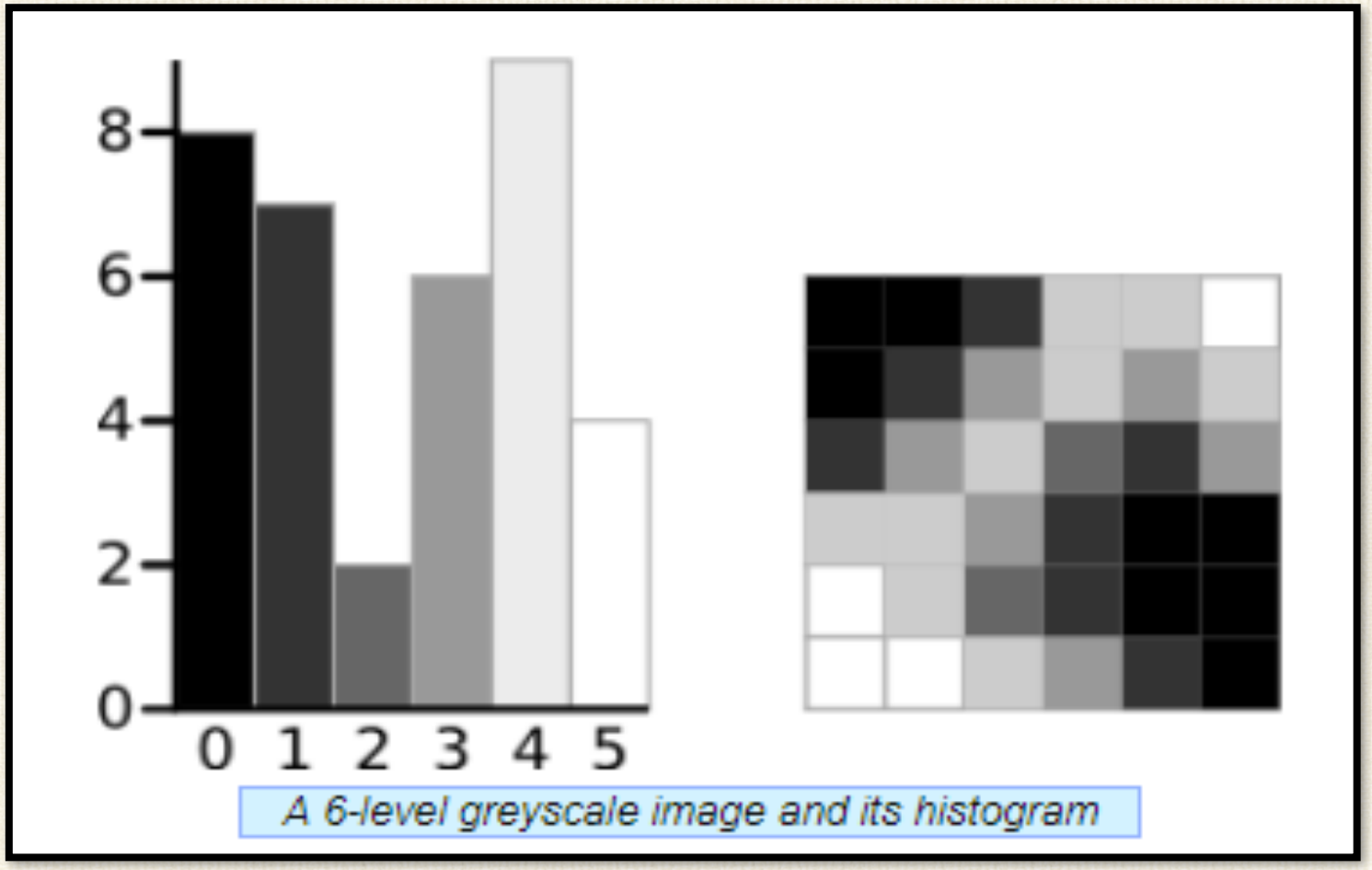



```java
public static long[] computeHist(int[][] img, int maxIntensity) {
    long[] hist = new long[maxIntensity];
    assert img.length > 0;
    int height = img.length;
    int width = img[0].length;
    for (int i = 0; i < height; i++){
        for(int j= 0; j < width; j++){
            int value = img[i][j];
            if((value >= 0)&&(value < maxIntensity)){
                hist[value]++;
            }
        }
    }
    return hist;
}
```

Otsu’s method - Demonstration

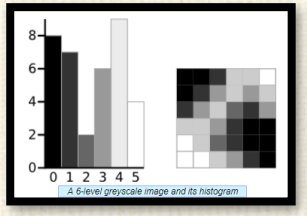
- Assumes a bimodal distribution of gray-level values

- Given 6x6 image



A 6-level greyscale image and its histogram

Otsu's method - Demonstration



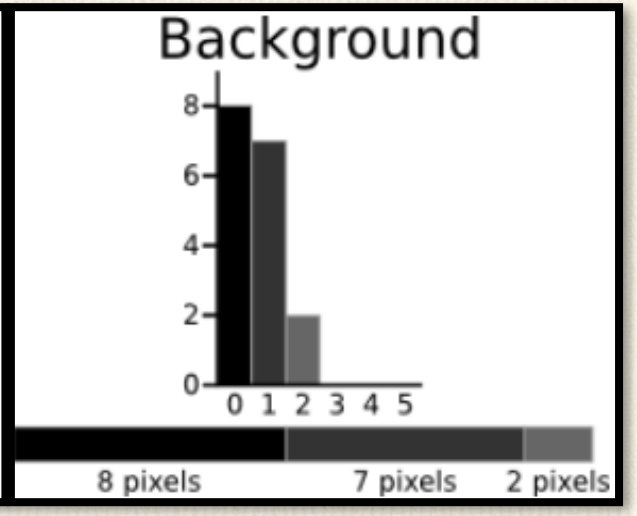
$$\text{Weight} \quad W_b = \frac{8+7+2}{36} = 0.4722$$

$$\text{Mean} \quad \mu_b = \frac{(0\times 8)+(1\times 7)+(2\times 2)}{17} = 0.6471$$

$$\text{Variance} \quad \sigma_b^2 = \frac{((0-0.6471)^2\times 8)+((1-0.6471)^2\times 7)+((2-0.6471)^2\times 2)}{17}$$

$$= \frac{(0.4187\times 8)+(0.1246\times 7)+(1.8304\times 2)}{17}$$

$$= 0.4637$$



Threshold = 3

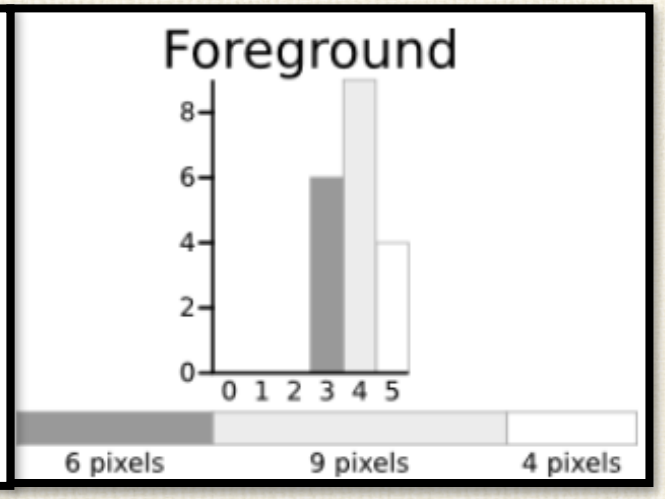
$$\text{Weight} \quad W_f = \frac{6+9+4}{36} = 0.5278$$

$$\text{Mean} \quad \mu_f = \frac{(3\times 6)+(4\times 9)+(5\times 4)}{19} = 3.8947$$

$$\text{Variance} \quad \sigma_f^2 = \frac{((3-3.8947)^2\times 6)+((4-3.8947)^2\times 9)+((5-3.8947)^2\times 4)}{19}$$

$$= \frac{(4.8033\times 6)+(0.0997\times 9)+(4.8864\times 4)}{19}$$

$$= 0.5152$$

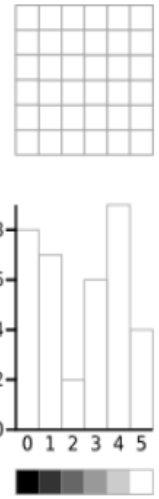
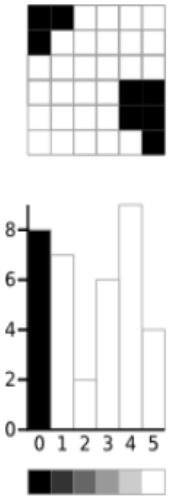
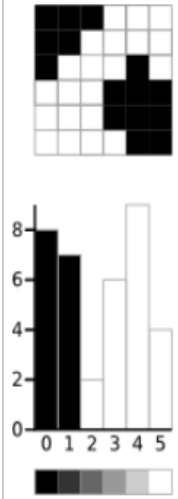
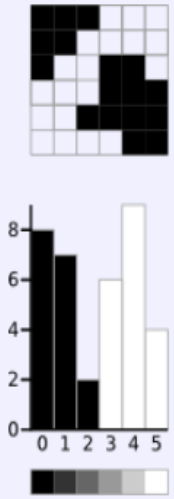
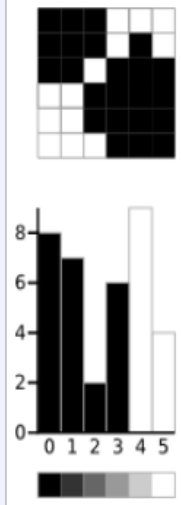
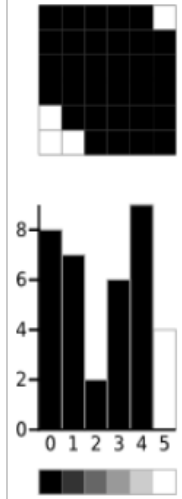


The next step is to calculate the 'Within-Class Variance'. This is simply the sum of the two variances multiplied by their associated weights.

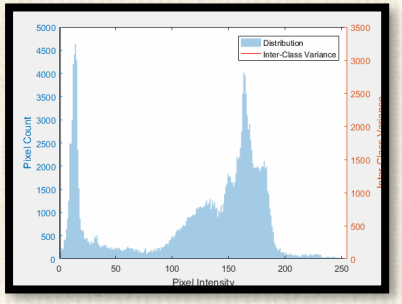
$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152 = 0.4909$$

Otsu's thresholding method involves iterating through all the possible threshold values .

The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum.

Threshold	T=0	T=1	T=2	T=3	T=4	T=5
						
Weight, Background	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Mean, Background	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Variance, Background	$\sigma^2_b = 0$	$\sigma^2_b = 0$	$\sigma^2_b = 0.2489$	$\sigma^2_b = 0.4637$	$\sigma^2_b = 1.4102$	$\sigma^2_b = 2.5303$
Weight, Foreground	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Mean, Foreground	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Variance, Foreground	$\sigma^2_f = 3.1196$	$\sigma^2_f = 1.9639$	$\sigma^2_f = 0.7755$	$\sigma^2_f = 0.5152$	$\sigma^2_f = 0.2130$	$\sigma^2_f = 0$
Within Class Variance	$\sigma^2_W = 3.1196$	$\sigma^2_W = 1.5268$	$\sigma^2_W = 0.5561$	**$\sigma^2_W = 0.4909$**	$\sigma^2_W = 0.9779$	$\sigma^2_W = 2.2491$

Otsu's method



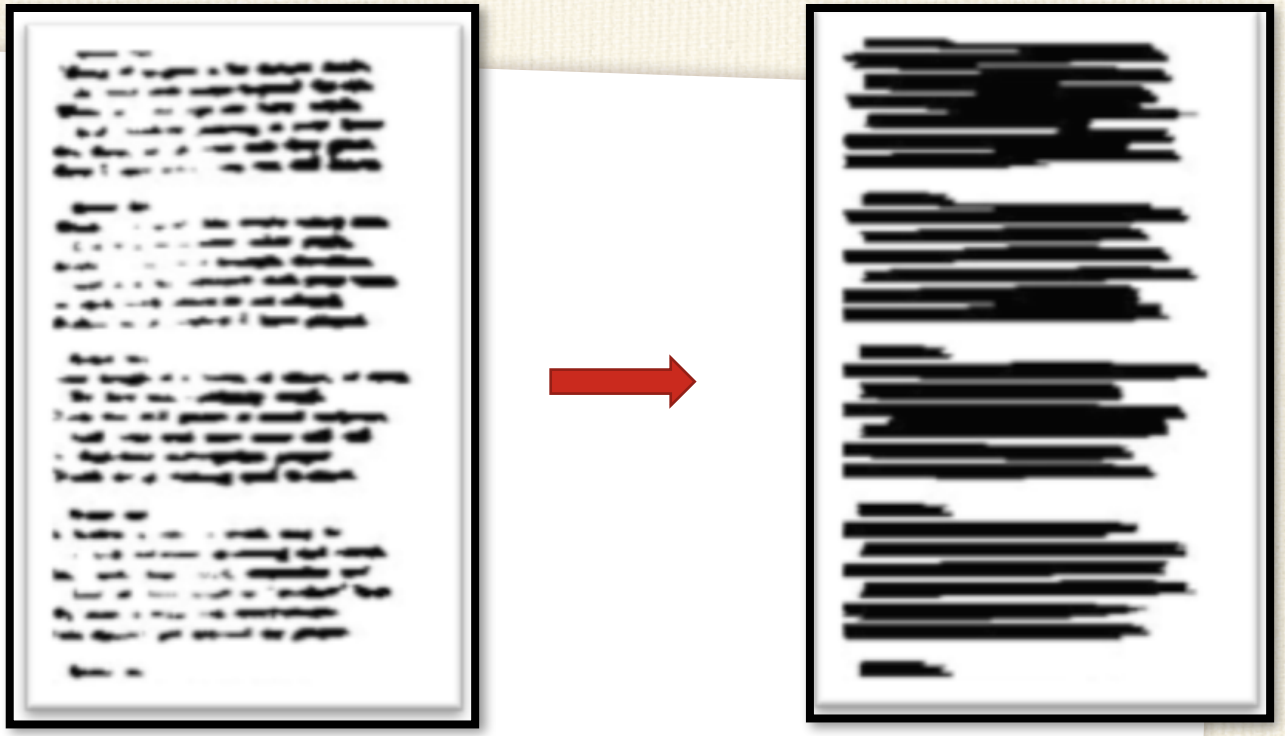
By a bit of manipulation, we can calculate what is called the *between class* variance, which is far quicker to calculate.

Luckily, the threshold with the maximum *between class* variance also has the minimum *within class* variance.

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2 \quad \text{(as seen above)}$$

$$\begin{aligned}\text{Between Class Variance} \quad \sigma_B^2 &= \sigma^2 - \sigma_W^2 \\ &= W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 \quad (\text{where } \mu = W_b\,\mu_b + W_f\,\mu_f) \\ &= W_b\,W_f\,(\mu_b - \mu_f)^2\end{aligned}$$

```
// Total number of pixels
int total = srcData.length;

float sum = 0;
for (int t=0 ; t<256 ; t++) sum += t * histData[t];

float sumB = 0;
int wB = 0;
int wF = 0;

float varMax = 0;
threshold = 0;

for (int t=0 ; t<256 ; t++) {
   wB += histData[t];               // Weight Background
   if (wB == 0) continue;

   wF = total - wB;                 // Weight Foreground
   if (wF == 0) break;

   sumB += (float) (t * histData[t]);

   float mB = sumB / wB;            // Mean Background
   float mF = (sum - sumB) / wF;    // Mean Foreground

   // Calculate Between Class Variance
   float varBetween = (float)wB * (float)wF * (mB - mF) * (mB - mF);

   // Check if new maximum found
   if (varBetween > varMax) {
      varMax = varBetween;
      threshold = t;
   }
}
```



Pixel consolidation

- Remove stray black spots are cleared.
- For each pixel in a row counts the total object pixels (black) in that row
- If the total number of object pixels in a row is greater than a given threshold, all of the pixels from the start index to the end index in the row are assigned to object pixels (Black)

Features extraction

- Computation of:
  - Column widths
  - Row depths
- Calculating statistics (mean, std, min, max, range) of:
  - Margin on the left
  - Jaggedness
  - Stanzas
  - Row lengths

Column widths computation

The algorithm counts both:

- length of background(white) pixels prior to the first object(black) pixel
- length of background pixels after the final object pixel in a row

```
public void computeColumnWidths()  {

    boolean stillBackground;
    int numPixSoFarFromLeft;
    int numPixSoFarFromRight;

    for (int i = 0; i < DEPTH; i++)  {

        // going from left to right
        numPixSoFarFromLeft = 0;
        stillBackground = true;
        int j = WOFFSET;
        while  (stillBackground && j < WIDTH - WOFFSET)  {
            if (image.getBinaryImagePixels()[i][j] == OBJECT)
                stillBackground = false;
            else
                numPixSoFarFromLeft++;
            j++;
        }

        // going from right to left
        numPixSoFarFromRight = 0;
        stillBackground = true;
        j = WIDTH - WOFFSET;
        while  (stillBackground && j >= WOFFSET)  {
            if (image.getBinaryImagePixels()[i][j] == OBJECT)
                stillBackground = false;
            else
                numPixSoFarFromRight++;
            j--;
        }

        leftColumnWidths[i] = numPixSoFarFromLeft;
        rightColumnWidths[i] = numPixSoFarFromRight;
    }

}  // end computeColumnWidths
```

* WIDTH = image.getHorizontal(); DEPTH = image.getVertical();
*WOFFSET = **(int) (WIDTH*0.1);*** *DOFFSET = **(int) (DEPTH*0.1);***

Row depths computation

The algorithm counts the continuous background(white) pixels in a each column and stores the values in a 2D integer matrix.

0
0
0
1
1
0
0
0
0
1
1
.

3
4
.
.
.

```
public void computeRowDepths()  {

    int numPixSoFar;
    int k;  // to hold the index to store each gap's depth
    for (int j = WOFFSET; j < WIDTH - WOFFSET; j++)  {
        k = 0;  // first index

        // going from top to bottom
        numPixSoFar = 0;
        int i = DOFFSET;
        while  (i < DEPTH - DOFFSET)  {
            if (image.getBinaryImagePixels()[i][j] == OBJECT){
                // we have found the first break of background pixels gap
                if (numPixSoFar > 0) {  // only update if we have accumulated
                    rowDepths[k][j] = numPixSoFar;                        //
                    numPixSoFar = 0; // reset for the next gap
                    k++;    // update index for the next gap
                }               }
            else
                numPixSoFar++;
            i++;
        }

    }  // end for j loop
}  // end computeRowDepths
```

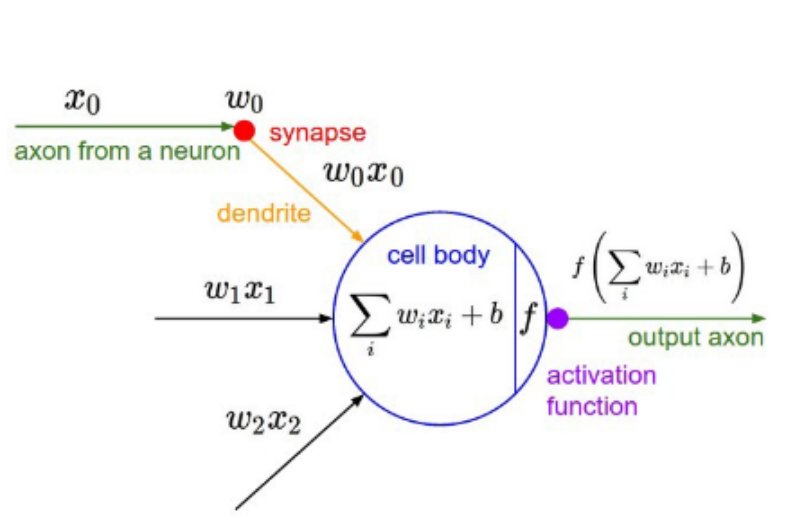
* WIDTH = image.getHorizontal(); DEPTH = image.getVertical();
*WOFFSET = **(int) (WIDTH*0.1);*** *DOFFSET = **(int) (DEPTH*0.1);***

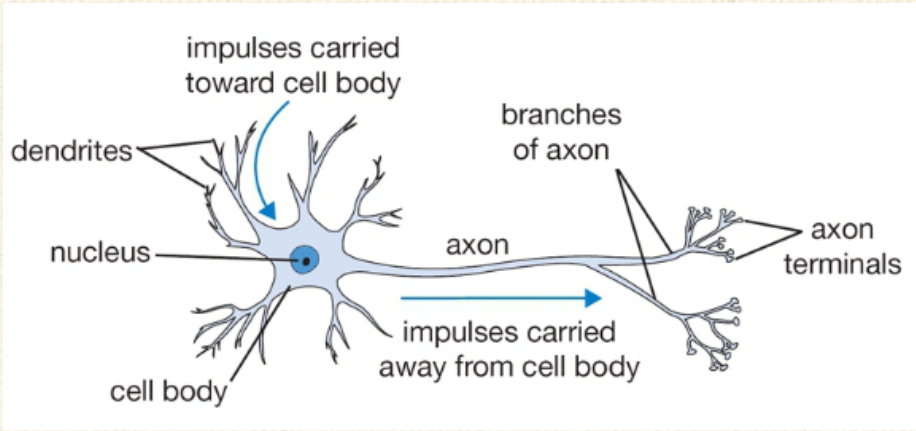
Calculating statistics (mean, std, min, max, range)

- Margin on the left - using the column widths on the left of each image
- Stanzas - looking for whitespace between stanzas, using row depths
- Jaggedness - measures of the background pixels after the final object pixel (using the column widths on the right of each image)
- Compute length of columns

ANN – Artificial Neural Network

- Inspired by the human brain
- The basic computational unit of the brain is a **neuron**.
- The node/neuron receives input from some other nodes and computes an output.
- Each input has an associated weight (w).
- The node applies a function to the weighted sum of its inputs.
- The idea is that the synaptic strengths (the weights *w*) are learnable and control the strength of influence.
- If the final sum is above a certain threshold, the neuron can *fire*, sending a spike along its axon.



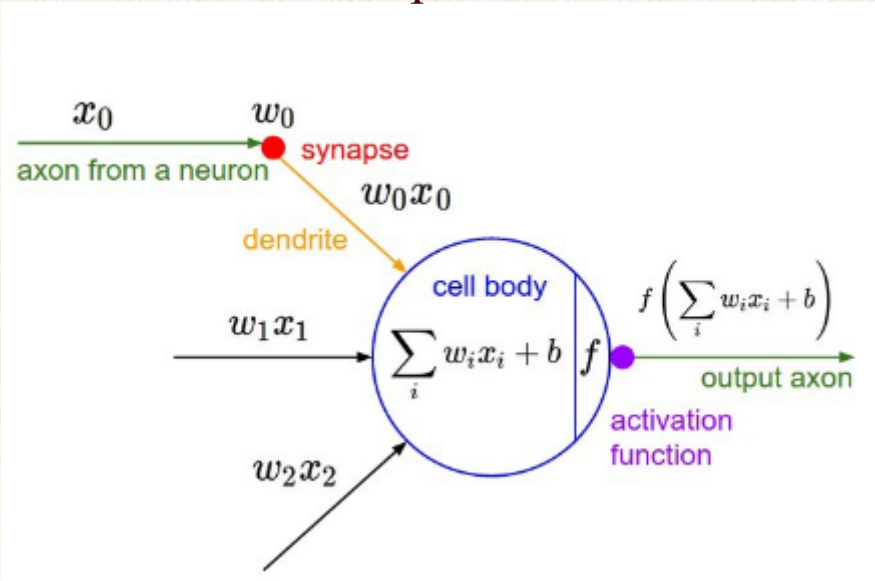


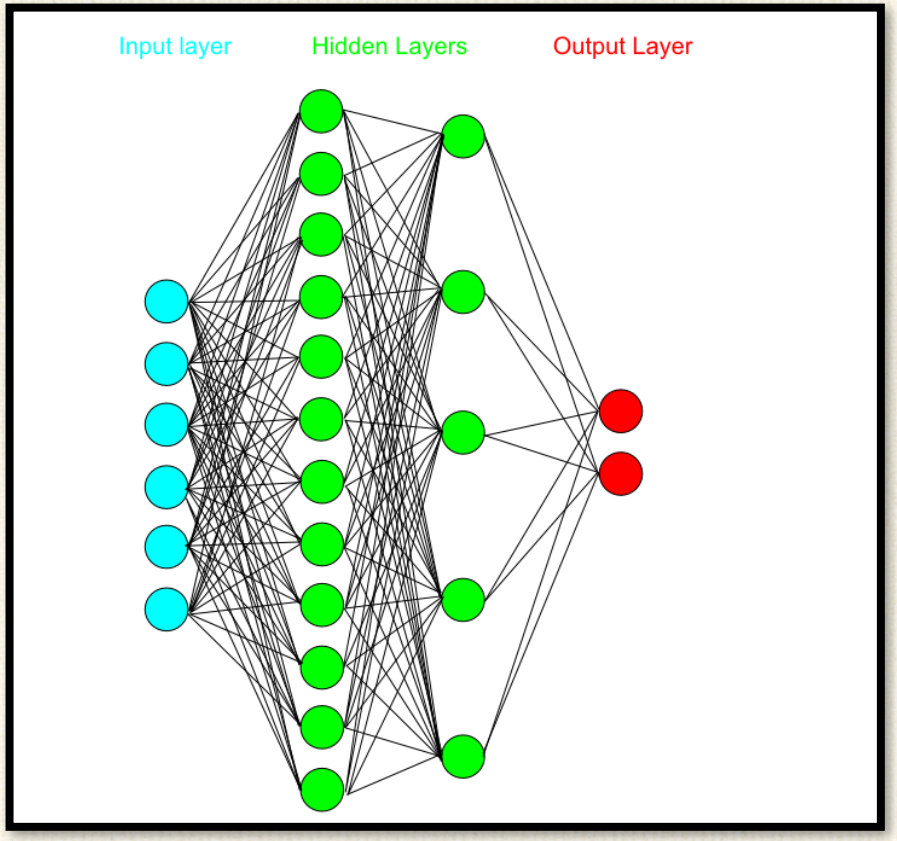
ANN – Multi-layer Perceptron

sigmoid function

$$g(x) = \frac{1}{1+e^{-x}}$$

- Consists of multiple layers of computational units
- Each neuron in one layer has directed connections to the neurons of the subsequent layer.
- Usually using sigmoid function as an activation function.
- MLP are able to learn non-linear representations





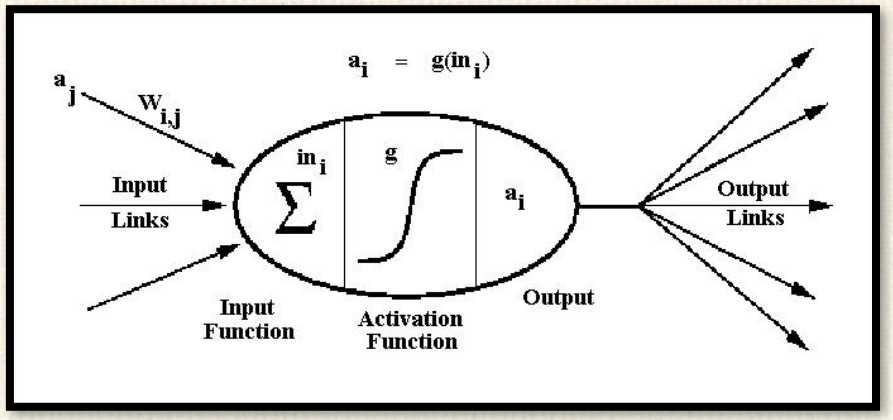
ANN – More details

Activation function - has to be a non-linear function, otherwise the neural network will only be able to learn linear models.

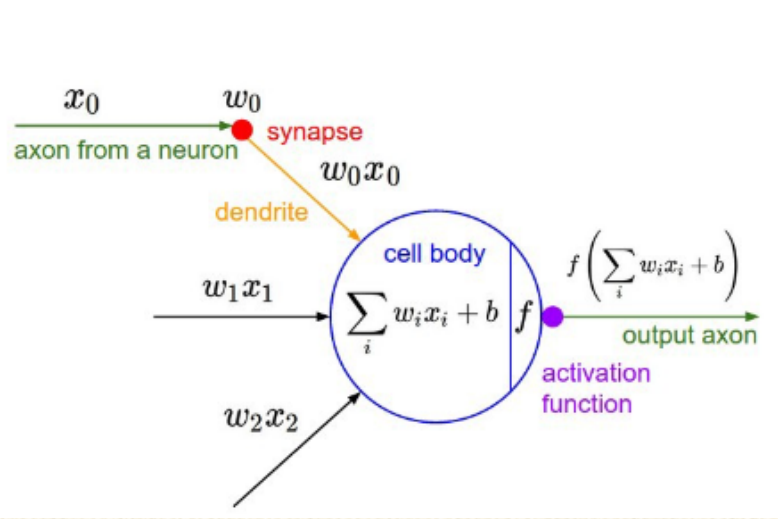
Error function - The goal is to learn the weights of the network automatically from data such that the predicted output y_{output} is close to the target y_{targe} for all inputs x_{input}. To measure how far we are from the goal, we use an error function

A commonly used error function is $E(y_{output}, y_{target}) = \frac{1}{2}(y_{output} - y_{target})^2$.

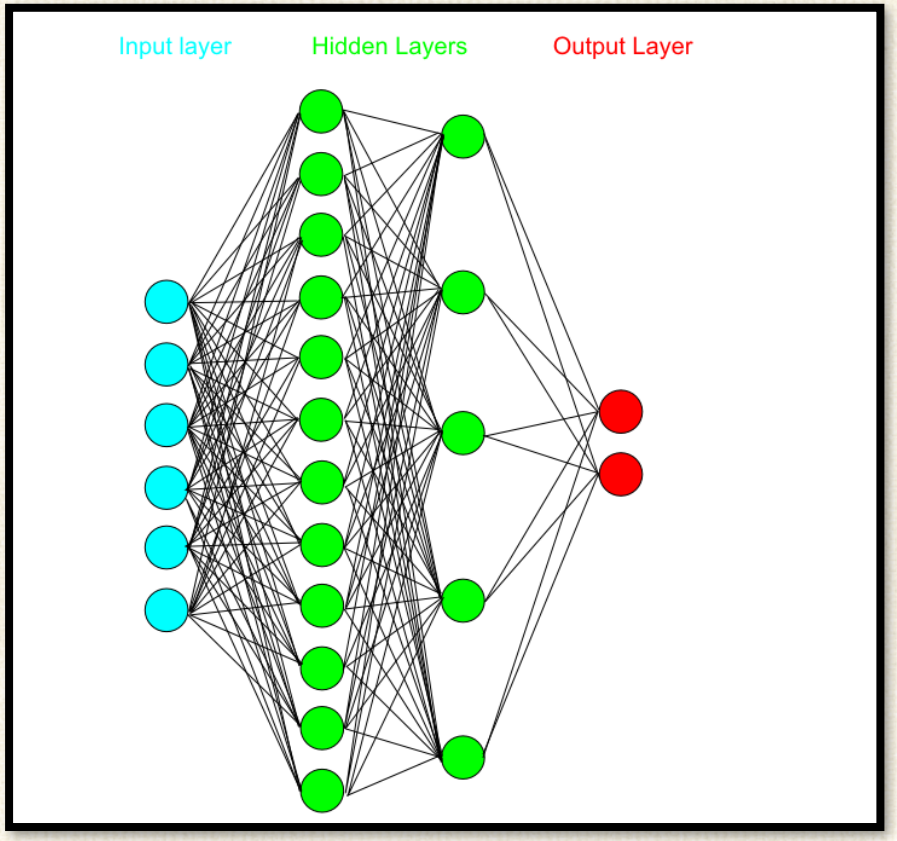
Backpropagation - Backpropagation minimizing the loss function, where the loss function determines how wrong the result is from what it's suppose to be.



ANN – In our case



- Attributes are translated into individual nodes that communicate with a hidden layer.
- These connections are initially weighted
- After a predetermined number of iterations the weights are increased or decreased depending on the prediction each node makes
- Over the iterations, the ANN is optimized such that the attributes that contribute most to determining the instance have the most weight, and through back propagation, the ANN reduces the weight of the less deterministic attributes.



Results

Class	Training	Testing
true & predicted true	79.44%	61.84%
true & predicted false	20.56%	36.18%
false & predicted true	8.26%	20.70%
false & predicted false	91.75%	79.30%
Precision		
Recall		

Recall =

Precision =

7: (a) the original image snippet, (b) the binary image, and (c) the consolidated binary in Our classifier mistakenly identified it as a poem image.

False positive – sample 1

☞Hugh Deary whose trial for murder
ok place at Hollidaysburg last week,
as been fully acquitted of the charge.

☞The alarm of fire on Tuesday eve-
ing was false.

MARRIED

On Monday afternoon, March 24th, by the
ev. L. Powell, Mr. John Bumgard to Miss
LEANOR WILLIAMS, all of this place.

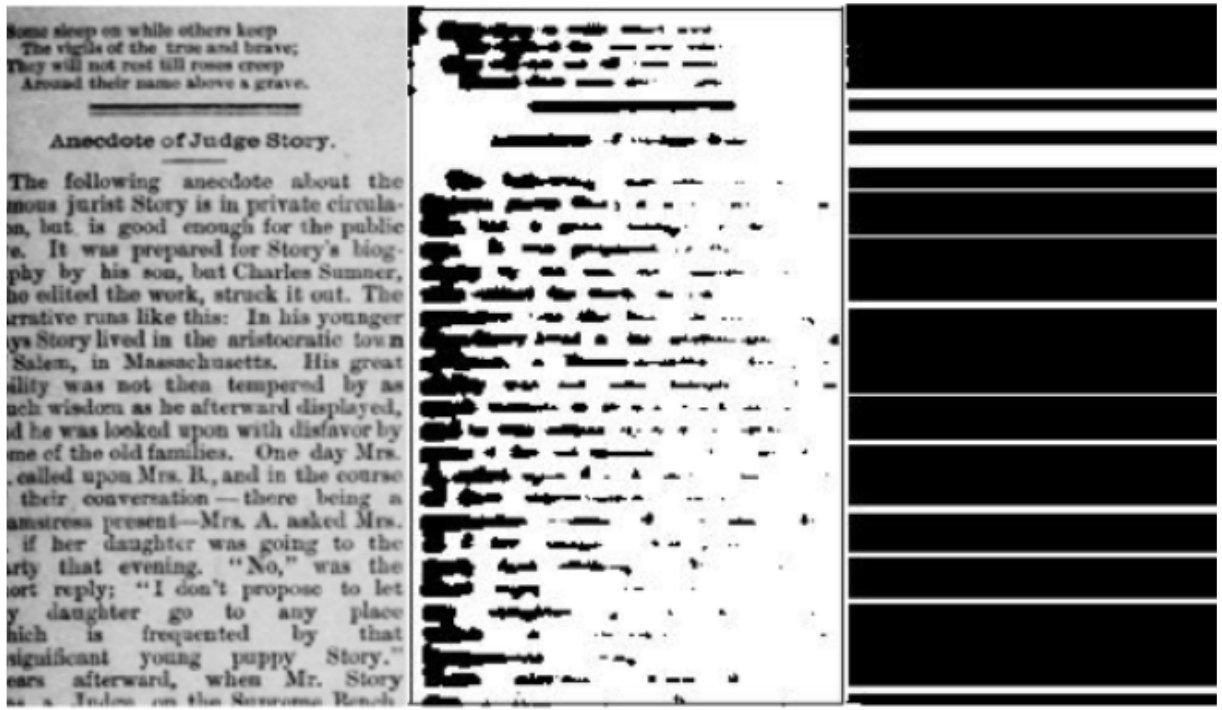
DIED

On Saturday morning, 15th inst., in White
wnship, after a short illness, Miss Lucy
NN, daughter of Samuel and Elizabeth Hol-
n, aged 18 years and 2 days.

The sudden demise of Miss Hollen has cast
shade of melancholy gloom over the minds
f a large number of attached relatives and
iends. She was confined to her room but a
w days, and during that time would insist
at she was recovering, even until within a
w hours of her death. Although it was evi-
ent to her friends that she was more seriously
disposed than she would admit, her death
as a surprise to all. But she died not as
hose having no hope; for her early connection
ith the Christian Church had enabled her to
ok upon death with no feelings of alarm, and
r parents and many relatives and friends

e 8: (a) the original image snippet, (b) the binary image, and (c) the consolidated binary ima
Our classifier mistakenly identified it as a poem image.

False positive – sample 2



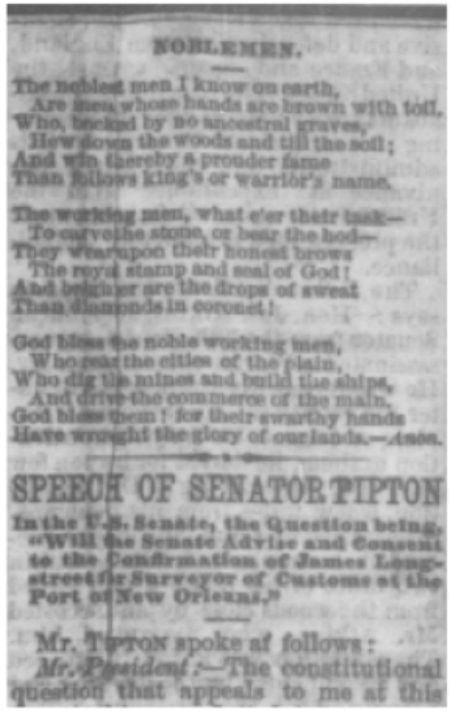
Some sleep on while others keep
The vigils of the true and brave;
They will not rest till roses creep
Around their name above a grave.

Anecdote of Judge Story.

The following anecdote about the
mous jurist Story is in private circula-
n, but is good enough for the public
e. It was prepared for Story's biog-
phy by his son, but Charles Sumner,
ho edited the work, struck it out. The
rrative runs like this: In his younger
ys Story lived in the aristocratic town
Salem, in Massachusetts. His great
ility was not then tempered by as
uch wisdom as he afterward displayed,
d he was looked upon with disfavor by
me of the old families. One day Mrs.
called upon Mrs. B., and in the course
their conversation — there being a
amstress present—Mrs. A. asked Mrs.
if her daughter was going to the
rty that evening. "No," was the
ort reply; "I don't propose to let
y daughter go to any place
hich is frequented by that
significant young puppy Story."
ears afterward, when Mr. Story

9: (a) the original image snippet, (b) the binary image, and (c) the consolidated binary i
Our classifier mistakenly identified it as a non-poem image.

False negative – sample 3



NOBLEMEN.

The noblest men I know on earth,
Are men whose hands are brown with toil,
Who, backed by no ancestral graves,
Hew down the woods and till the soil;
And win thereby a prouder fame
Than follows king's or warrior's name.

The working men, what e'er their task—
To carve the stone, or bear the hod—
They wear upon their honest brows
The royal stamp and seal of God!
And brighter are the drops of sweat
Than diamonds in coronet!

God bless the noble working men,
Who rear the cities of the plain,
Who dig the mines and build the ships,
And drive the commerce of the main.
God bless them! for their swarthy hands
Have wrought the glory of our lands.—*Anon.*

SPEECH OF SENATOR TIPTON

In the U.S. Senate, the Question being, "Will the Senate Advise and Consent to the Confirmation of James Longstreet for Surveyor of Customs of the Port of New Orleans?"

Mr. TIPTON spoke as follows:

Mr. President:—The constitutional question that appeals to me at this

Figure 10: The original image snippet with significant bleed-through.
ır algorithm failed to identify a viable threshold to classify the image into object and background pixels.
Our classifier mistakenly identified the snippet as a non-poem image.

False negative – sample 4

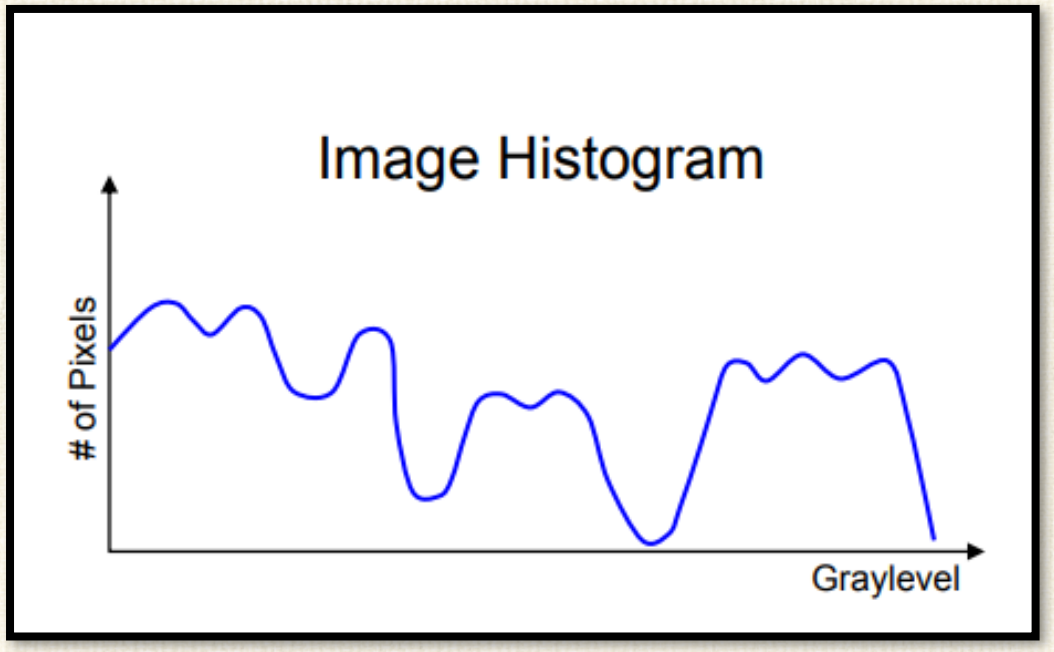
Improvements

- Improve extraction algorithms (for example binarization)
- Page segmentation
- More visual features
- Enlarge scaling

Simple thresholding is not always possible:

Back to binarization

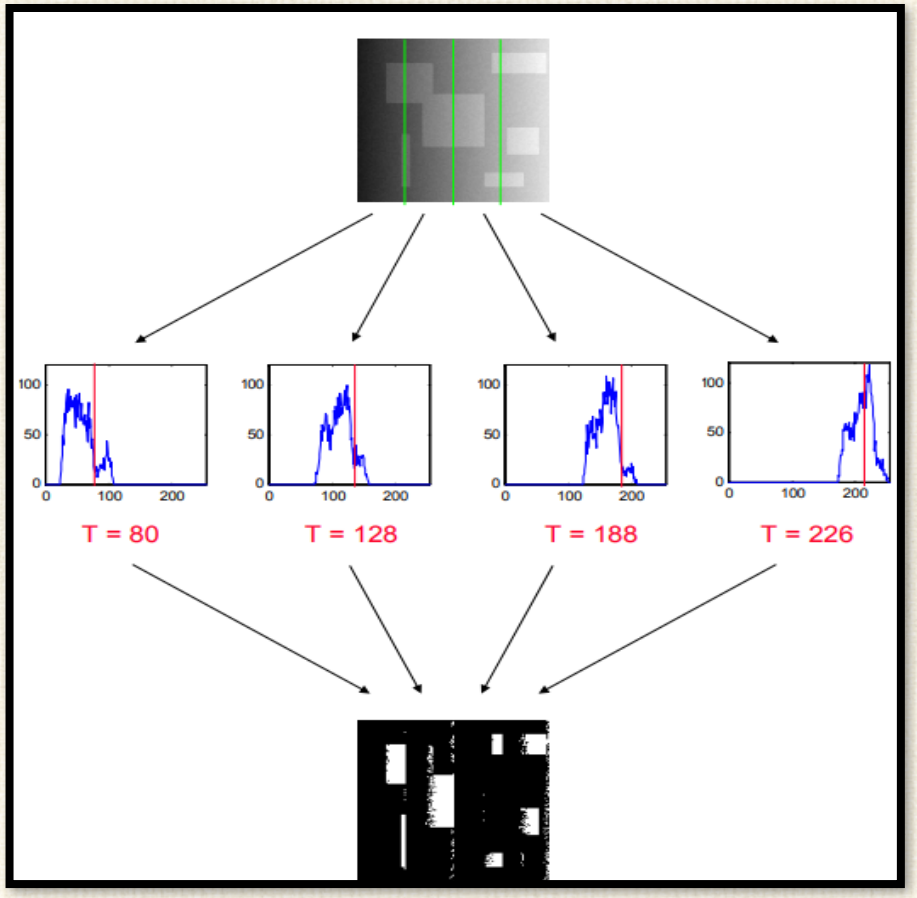
- Many objects at different gray levels.
- Variations in background gray level.
- Noise in image.



Local Thresholding - 4 Thresholds

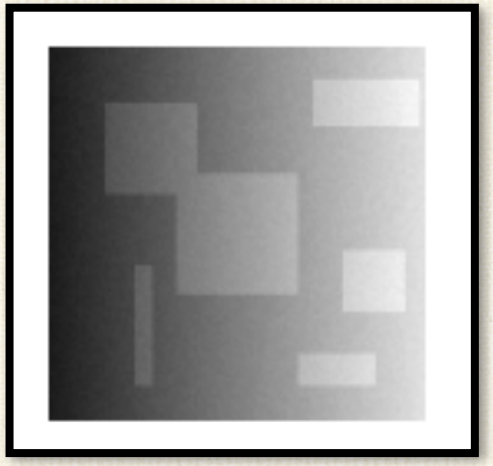
Divide image in to regions.

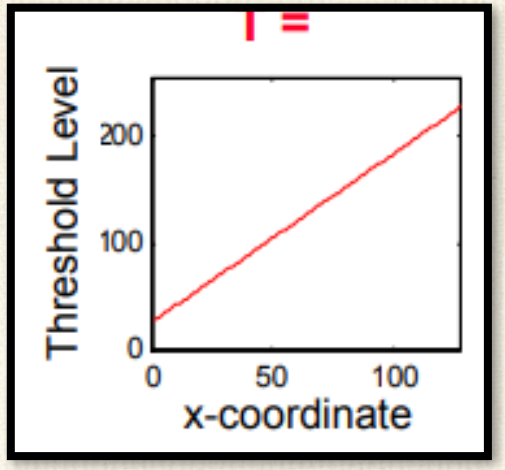
Perform thresholding independently in each region.



Adaptive Thresholding

Every pixel in image is thresholded according to the histogram of the pixel neighborhood.





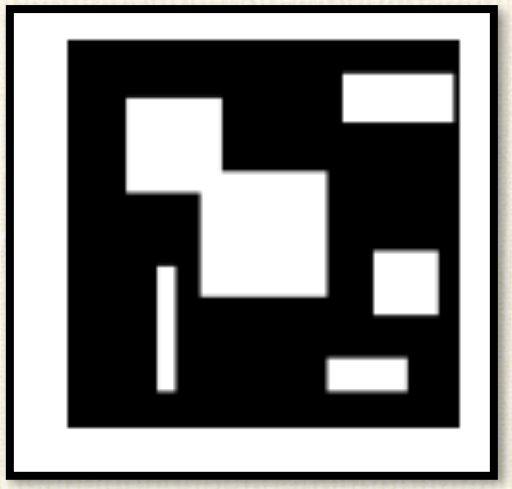


Image segmentation

Image segmentation is defined as a process of partitioning a digital image into multiple smaller segments called regions

Figure 11: Red lines for the column breaks

Image segmentation

Algorithm: Page Segmentation

Input: an original image, $I_{original}$, of a newspaper page

Output: a set of image snippets, $\langle i_{original} \rangle$

1. Compute average intensity of $I_{original}$,

2. If *AveIntensity*($I_{original}$) is too bright then
   a. Perform contrast enhancement on $I_{original}$ to obtain $I_{enhanced}$

3. Perform binarization on $I_{enhanced}$ to obtain I_{binary}

4. Perform morphological cleaning on I_{binary} to obtain $I_{binary\_cleaned}$ to clean up image noise

5. ColumnBreaks ← *FindColumnBreaks*($I_{binary\_cleaned}$)

6. $\langle i_{original} \rangle$ ← *GenerateSnippets*(ColumnBreaks, $I_{original}$)

Conclusion & Final thoughts