

January 6, 2020

# Image Recognition

**Goal:** Identify the word in the picture and use it to generate the card.

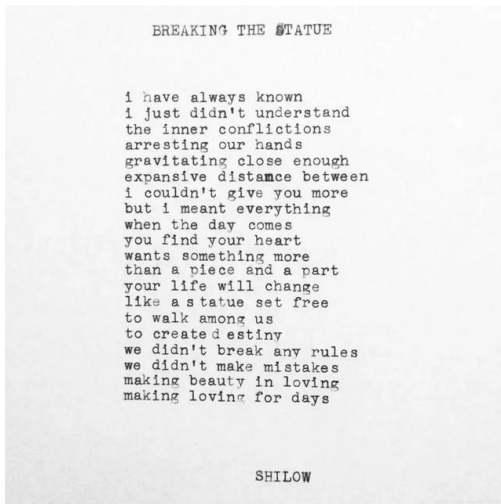
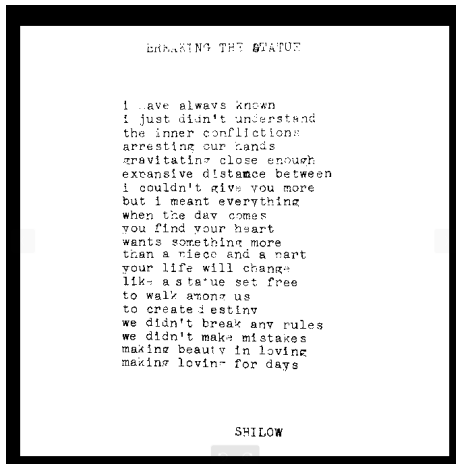


Figure: Target picture

# Method

**Step1:** Convert the picture to gray color. Then, set the blacker one to total black, whiter one to total white.

**Step1-1:** To realize step1, we would turn the picture into array to deal with it.



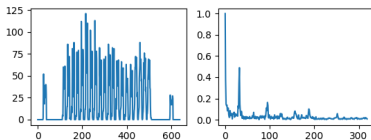
# Task

## How computer see a word?



# Fourier Transform

It seems good but awful.

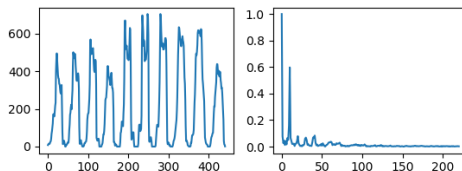


## BREAKING THE STATUE

i have always known  
i just didn't understand  
the inner conflixtions  
arresting our hands  
gravitating close enough  
expansive distance between  
i couldn't give you more  
but i meant everything  
when the day comes  
you find your heart  
wants something more  
than a piece and a part  
your life will change  
like a statue set free  
to walk among us  
to create destiny  
we didn't break any rules  
we didn't make mistakes

# Fourier Transform

Fewer data lead to less precision.



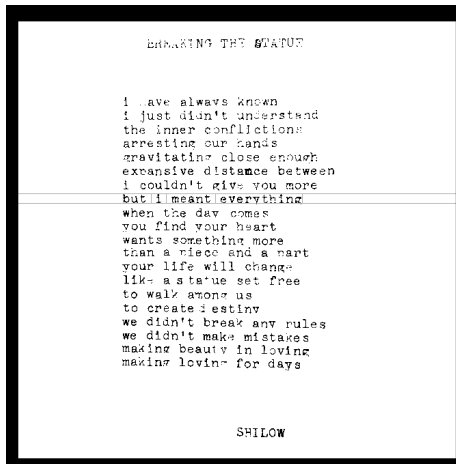
Here, then, is a wonderful machine for generating new solutions, with higher and lower energies—if we could just find *one* solution to get started! We call  $\hat{a}$  **ladder operators**, because they allow us to climb up and down in energy;  $\hat{a}_+$  is the **raising operator**, and  $\hat{a}_-$  the **lowering operator**. The “ladder” of states is illustrated in Figure 2.5.

But wait! What if I apply the lowering operator repeatedly? Eventually I’m going to reach a state with energy less than zero, which (according to the general theorem in Problem 2.3) does not exist! At some point the machine must fail. How can that happen? We know that  $\hat{a}_- \psi$  is a new solution to the Schrödinger equation, but *there is no guarantee that it will be normalizable*—it might be zero, or its square integral might be infinite. In practice it is the former: There occurs a “lowest rung” (call it  $\psi_0$ ) such that

# Method

**Step2-1:** When the black pixels number is below the threshold number, recognize it as a white row. This can help us detect the line.

**Step2-2:** Analyze the line, detect the wider white column to recognize the word.



# Method

**Step3-1:** Detect the words in the line, use the order to determine which word I selected. We get WordFromLine.

**Step3-2:** Detect the word in the region I selected. We get WordFromWord.

**Step3-3:** If the WordFromLine is similar to WordFromWord or  $\text{len}(\text{WordFromWord})=0$ , return WordFromLine. Else, return WordFromWord.

