In the Name of God

Signals and Systems

Project Phase 2

Spring Semester 1403-04
Department of Electrical Engineering

Sharif University of Technology
Instructor: Dr. B. Khalaj

Due Date: 1404/04/13

Project members:

Seyed Mohammad Hasan Mirshafiei 402102551

Mohammad Javad Yousefi 402102776

- Project Report -

1. Theoretical Questions

1.1.1 Two-Dimensional Correlation

In this project, you will perform character recognition using correlation. Although correlation is a technique traditionally used for comparing one-dimensional signals, it can also be adapted for image processing tasks such as template matching.

Your goal is to investigate how this one-dimensional correlation method can be used to compare images or parts of images (e.g., character segments). This approach is at the heart of the recognition method used in this project.

1.1.1 Questions

1. What is the mathematical definition of correlation between two signals? Explain it in your own words and write the standard formula. What does the correlation value represent? (5)

- Correlation is a mathematical operation that measures the similarity between two signals. It functions by sliding one signal across another and calculating a "similarity score" at each position. The position where this score is maximized indicates the point of greatest alignment between the two signals. This process allows us to determine if a specific, smaller pattern is "hidden" within a larger signal and, if so, to locate its precise position.

- **Mathematical Formula**

  For two discrete signals, $f[n]$ and $g[n]$, the cross-correlation, denoted by the symbol $\star$, is defined as:

$$(f \star g)[n] = \sum_{k=-\infty}^{\infty} f[k] \cdot g[n+k]$$

  In this formula, n represents the amount of shift, or "lag," between the two signals.

- **An Example:**

  Let's assume we have a $\boldsymbol{long\ signal}\ (\boldsymbol{f})$ and a $\boldsymbol{short\ pattern}\ (\boldsymbol{g})$, and our goal is to find where the pattern g is located within the main signal f.

$$\boldsymbol{Main\ Signal\ f}\text{: } [2, 1, 8, 9, 3, 1, 0]$$

$$\boldsymbol{Pattern\ to\ Find\ g}\text{: } [8, 9, 3]$$

  By calculating the correlation for different shift values (n), we will search for the point where the similarity score is maximized.

  Calculation for Several Shift Values (n):

  Case 1: $\boldsymbol{n = 0}\ (\boldsymbol{No\ Shift})$

$$Score = (2 \times 8) + (1 \times 9) + (8 \times 3) = 16 + 9 + 24 = 49$$

  Case 2: $\boldsymbol{n = 1}\ (\boldsymbol{Pattern\ shifted\ one\ unit\ to\ the\ right})$

$$Score = (1 \times 8) + (8 \times 9) + (9 \times 3) = 8 + 72 + 27 = 107$$

  Case 3: $\boldsymbol{n = 2}$

$$Score = (8 \times 8) + (9 \times 9) + (3 \times 3) = 64 + 81 + 9 = 154$$

  Case 4: $\boldsymbol{n = 3}$

$$Score = (9 \times 8) + (3 \times 9) + (1 \times 3) = 72 + 27 + 3 = 102$$

  By reviewing the scores, we observe that the maximum value (the peak) is **154**, which occurs at a shift of $\boldsymbol{n = 2}$.

- **What the Correlation Value Represents**

  The output of these calculations provides the following information:

  Peak of the Output Signal: The maximum correlation value (in our example, **154**) represents the degree of similarity. The larger this number, the more closely the pattern matches that segment of the signal.

  Position of the Peak: The position where the peak occurs (in our example, **n=2**) tells us that the best alignment is achieved by shifting the pattern g by 2 units to the right. This result correctly shows us that the pattern [8, 9, 3] begins at the third element (index 2) of the signal f.

---

2. Research and explain how a method designed for one-dimensional signals can be used to compare images or image segments. Describe your proposed approach clearly. (Do not explain 2D correlation.) (2.5)

- To apply a one-dimensional correlation method for comparing images (which are two-dimensional structures), we must first transform the 2D image data into a 1D signal. The proposed approach for this is a process called **"Flattening"**.

- **Proposed Approach: Image Flattening**

  This approach consists of the following steps:

  1. **Transforming the Image into a 1D Signal:** A 2D image is essentially a matrix of pixels. To convert it into a 1D signal, we proceed row by row:
     - First, we take all the pixels from the **first row** of the image and place them sequentially in a 1D array.
     - Next, we take the pixels from the **second row** and append them to the end of the array created in the previous step.
     - This process continues for all rows until the very last row of the image.

     At the end of this process, we will have a single, long vector (or 1D array) that contains all the pixel information from the 2D image.

  2. **Applying 1D Correlation:** Now that both the **image segment (e.g., a segmented character)** and the **reference template** have been converted into 1D signals, we can directly apply the standard 1D correlation formula (as described in Question 1) to them.

3. **The Comparison Process:** To recognize a character, we first flatten the segmented character image into a vector. Then, we flatten each reference template (e.g., for numbers 0-9) and compute the 1D correlation between the character's vector and each template's vector. The template that produces the **highest correlation score** is chosen as the final recognition result.

1.1.2 Image Blurring and Degradation Modeling

Blurring in images often results from motion, out-of-focus lenses, or deliberate smoothing filters. In signal processing terms, this can be modeled as the output of a system applied to an image. For example:

$H_{\text{blur}}(z) = \frac{1-p}{1-pz^{-1}}$

This is a recursive 1D IIR filter (Infinite Impulse Response), often used for exponential smoothing or blurring.

1.1.2 Questions

1. Explain how this 1D filter is applied to images, and also explain the parameter p and its effect on the filter's performance. (5)

In image processing, 2D images are typically represented as matrices with rows and columns of pixel values. s. To apply $H_{blur}(Z)$ a 1D filter like to an image, we process the image one dimension at a time. either along the rows (horizontally) or along the columns (vertically).

the filter's operation can be understood through its difference equation:

$$(1 - pz^{-1})Y = (1 - p)X$$

$$Y[n] - pY[n-1] = (1-p)X[n]]$$

For each row or column, the filter starts with an initial condition: $Y(-1) = 0$

and computes $y[n]$ sequentially for $n = 0,1,2, \ldots$

For:

$$0 < p < 1$$

$$h[n] = \left(\frac{p-1}{p}\right)p^n u[n]$$

P^n is a decaying exponential:

- If p is close to 0: P^n decays quickly so $h[n]$ has a large negative initial value and fades rapidly, implying minimal blurring
- If p is close to 1: P^n decays slowly so $h[n]$ has a smaller negative amplitude but persists longer, increasing the blurring

---

2. If we assume that we have the blur transformation function (blur kernel) of the channel, can we recover a high-quality and clean image? Why? (2.5)

To invert the blurred image to the original image we should H_{blur}^{-1} apply to the blurred image.

We can show that $H_{blur}(Z)$ the is Invertible so we can reconstruct the original image

$$\frac{1}{H_{bure}} = \frac{1 - pz^{-1}}{1 - p^{-1}}$$

Cause of that the denominator is zero only at p=1 this is invertible.

1.1.3 Sampling Rate and Resolution Effects

In image processing, sampling corresponds to the image resolution. Reducing the resolution is equivalent to downsampling the image, which affects the frequency content and the ability to extract fine details such as the shape of characters. If sampling is too coarse, important features may be lost. In this project, we test how much the resolution can be reduced while still enabling character recognition via correlation.

1.1.3 Questions

1. Do a bit of research on 2D sampling theory and briefly summarize your understanding.(5)

   Sampling and Resolution

   In image processing, sampling is the process of converting a continuous, analog image (like a real-world scene) into a discrete, digital format (a set of pixels). This is achieved by measuring the brightness values at regular points on a grid.

   The distance between these points defines the sampling rate, which is directly related to the image's resolution. The closer the points are, the higher the sampling rate and, consequently, the higher the resolution, allowing for finer details to be captured.

The Nyquist Sampling Theorem

This fundamental theorem states that to perfectly reconstruct an image from its samples without losing information, the sampling rate must be at least twice the highest frequency present in the image.

- **High frequencies** in an image represent areas with rapid changes, such as sharp edges, fine lines, and complex textures. The distinct shape of a license plate character is composed of high-frequency information.
- **Low frequencies** represent smooth and uniform areas of an image.

Aliasing: The Consequence of Insufficient Sampling

If the sampling rate is less than the Nyquist rate (a condition known as **undersampling**), an artifact called **aliasing** occurs. In this phenomenon, high-frequency information is incorrectly interpreted as lower frequencies. In practice, this error manifests as **jagged, "stair-step" edges** on what should be smooth, diagonal lines.

'Connection to the Project'

When we **downsample** the license plate image, we are reducing its sampling rate. As long as this rate remains high enough, the sharp, high-frequency details of the characters are preserved. However, when the resolution is reduced too much, aliasing degrades the precise shape of the characters. This loss of critical feature information is the primary reason why the correlation-based recognition algorithm fails at low sampling rates.

2. Implementation

"Ideal case"

Part I & II: Plate Segmentation and Recognition (*ideal.py*)

The initial phase of this project is divided into two primary parts: first, analyzing an ideal license plate image to segment it into its constituent characters, and second, recognizing each character to reconstruct the full plate number. This process is executed through the following three stages:

1. **Preprocessing**: Preparing the image for analysis.

2. **Character Segmentation**: Finding and isolating each individual character on the plate.
3. **Character Recognition**: Identifying each segmented character using template matching.

---

1. Image Preprocessing

Cropping:

Given that the input image is considered ideal and static, the first step is to isolate the license plate area using predefined pixel coordinates. This step defines the Region of Interest (ROI), which ensures that subsequent processing is focused, efficient, and less prone to errors from background noise.

Thresholding:

The cropped image is first converted to grayscale. Subsequently, a fixed threshold is applied to convert it into a binary (pure black and white) image. The THRESH_BINARY_INV mode was used specifically to render the characters in white (pixel value 255) and the background in black (pixel value 0), which is the ideal format for contour detection.

---

2. Character Segmentation

Contour Detection

Using the cv2.findContours function, the boundaries of all distinct white objects (the characters) in the binary image are identified.

Contour Filtering

As minor noise or imperfections might also be detected as contours, a filtering process is applied based on geometric properties. Only contours whose height and aspect ratio fall within the expected range of a standard character are retained; all others are discarded.

Sorting

The remaining valid contours are sorted based on their x-axis position. This ensures the characters are processed in the correct left-to-right order, which is essential for reconstructing the final plate string.

Standardization (Padding)

To improve the accuracy of the template matching stage, a fixed **5-pixel border (padding)** is added to all sides of each segmented character. This standardization step ensures that the segmented characters share a similar framing and aspect ratio to the reference templates, which is crucial for an accurate comparison.

```
Step 1: Segmenting characters from the license plate...
Segmented characters saved successfully!
Saving location: c:\Users\mirsh\Desktop\S&S_Project\segmented_characters
```

---

3. Character Recognition

Template Matching

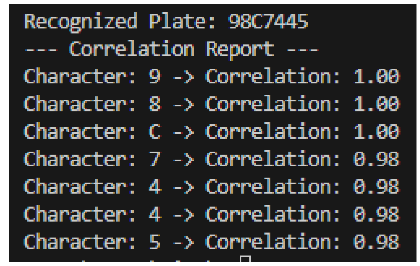
The core of the recognition logic resides in this section. The system iterates through each segmented character and compares it against a predefined library of reference templates. These templates are located in the numbers folder and contain high-quality images of numbers (0-9) as well as letters (A-D).

Correlation Coefficient

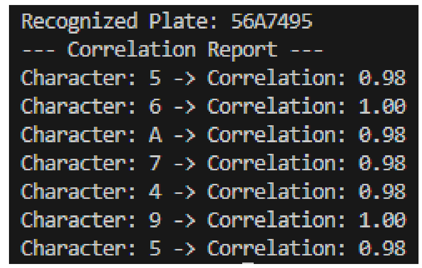
The comparison is performed using the **Normalized Cross-Correlation Coefficient** (TM_CCOEFF_NORMED) as the similarity metric. This method returns a score between -1.0 and +1.0, where a higher value indicates a greater similarity. The template that yields the highest correlation score is selected as the correct identification for the character.

```
Step 2: Recognizing segmented characters...

Recognized Plate: 93D4328
--- Correlation Report ---
Character: 9 -> Correlation: 1.00
Character: 3 -> Correlation: 1.00
Character: D -> Correlation: 1.00
Character: 4 -> Correlation: 0.98
Character: 3 -> Correlation: 1.00
Character: 2 -> Correlation: 1.00
Character: 8 -> Correlation: 1.00
```

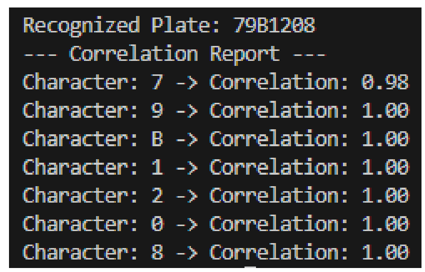
(This section output of the rest of the license plates of the cars)



```
Recognized Plate: 98C7445
--- Correlation Report ---
Character: 9 -> Correlation: 1.00
Character: 8 -> Correlation: 1.00
Character: C -> Correlation: 1.00
Character: 7 -> Correlation: 0.98
Character: 4 -> Correlation: 0.98
Character: 4 -> Correlation: 0.98
Character: 5 -> Correlation: 0.98
```



```
Recognized Plate: 56A7495
--- Correlation Report ---
Character: 5 -> Correlation: 0.98
Character: 6 -> Correlation: 1.00
Character: A -> Correlation: 0.98
Character: 7 -> Correlation: 0.98
Character: 4 -> Correlation: 0.98
Character: 9 -> Correlation: 1.00
Character: 5 -> Correlation: 0.98
```



```
Recognized Plate: 79B1208
--- Correlation Report ---
Character: 7 -> Correlation: 0.98
Character: 9 -> Correlation: 1.00
Character: B -> Correlation: 1.00
Character: 1 -> Correlation: 1.00
Character: 2 -> Correlation: 1.00
Character: 0 -> Correlation: 1.00
Character: 8 -> Correlation: 1.00
```

Part III: Analysis of the Robustness of a License Plate Recognition Algorithm Against Image Quality Reduction Using the Subsampling Method (*ideal_resize_analysis.py*)

This section of the project achieves two primary objectives, which are detailed below:

1. A comparison between two down-sampling methods: **Subsampling** and interpolation-based **resizing** using the OpenCV library.
2. An analysis of license plate recognition at various down-sampling rates to identify the algorithm's **breaking point**.

Subsampling Technique

To simulate a lower-resolution input, we employed the **subsampling** technique using NumPy's array slicing (image[::rate, ::rate]). In this method, for a given rate N, only one pixel is selected from every N pixels in the rows and columns; the rest are completely discarded. Unlike methods such as OpenCV's resize function, this approach does not perform any averaging or interpolation, meaning it does not generate new pixel values. This technique provides a pure model of information loss, similar to capturing an image with a lower-resolution camera sensor.

Character Segmentation and Recognition Pipeline

After down-sampling the license plate image in the initial step, the following pipeline is executed to segment and recognize the characters:

- **Character Segmentation**: The license plate area is first cropped from the smaller, down-sampled image. Then, using image processing techniques such as binary **thresholding** and **contour detection**, each character is isolated from the background.
- **Dynamic Padding**: To standardize the segmented characters for comparison against the reference templates, a **dynamic border (padding)** is added. The size of this padding is scaled proportionally to the down-sampling factor, starting from a base of 5 pixels on each side for the original, full-resolution image. This ensures the character's aspect ratio is not distorted.
- **Template Matching**: Finally, each segmented character is compared against a set of high-resolution reference templates. To perform the comparison, the high-resolution template is resized down to the exact dimensions of the small character using the cv2.resize function. The **correlation coefficient** between the two images is then calculated, and the template with the highest score is selected as the recognized character.

Quantitative Results

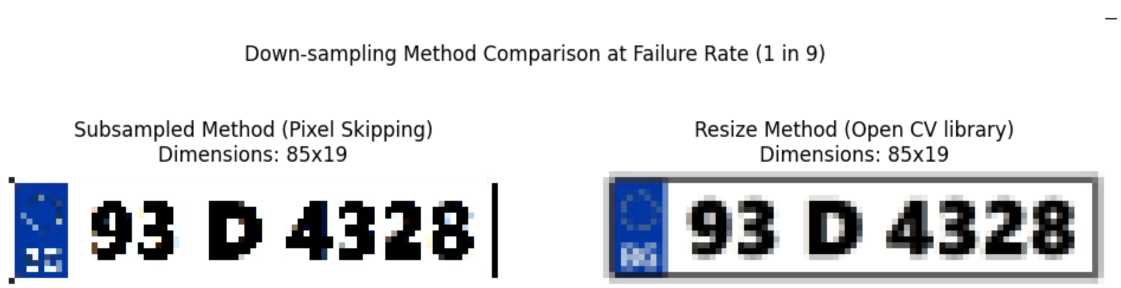
The following table summarizes the system's performance at various subsampling rates. As observed, with an increasing rate N (and decreasing quality), the average correlation coefficient gradually decreases.

```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '93D4328'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)    | Recognized Plate      | Avg. Correlation      | Correct?
--------------------------------------------------------------------------
1                | 93D4328               | 0.9966                | Yes
2                | 93D4328               | 0.9517                | Yes
3                | 93D4328               | 0.8485                | Yes
4                | 93D4328               | 0.9281                | Yes
5                | 93D4328               | 0.9708                | Yes
6                | 93D4328               | 0.9024                | Yes
7                | 93D4328               | 0.8732                | Yes
8                | 93D4328               | 0.8162                | Yes
9                | 93D432B               | 0.8574                | No
10               | 93D4320               | 0.8449                | No
11               | 93D4328               | 0.7919                | Yes
12               | 9304188               | 0.7170                | No
13               | 93043                 | 0.7002                | No
14               | 9304336               | 0.5994                | No
15               | 91026                 | 0.5917                | No
16               | 02A                   | 0.6374                | No
17               | A388                  | 0.5997                | No
18               | 916417A               | 0.5260                | No
19               | 08A2A                 | 0.5215                | No
20               | A503A                 | 0.5015                | No
```

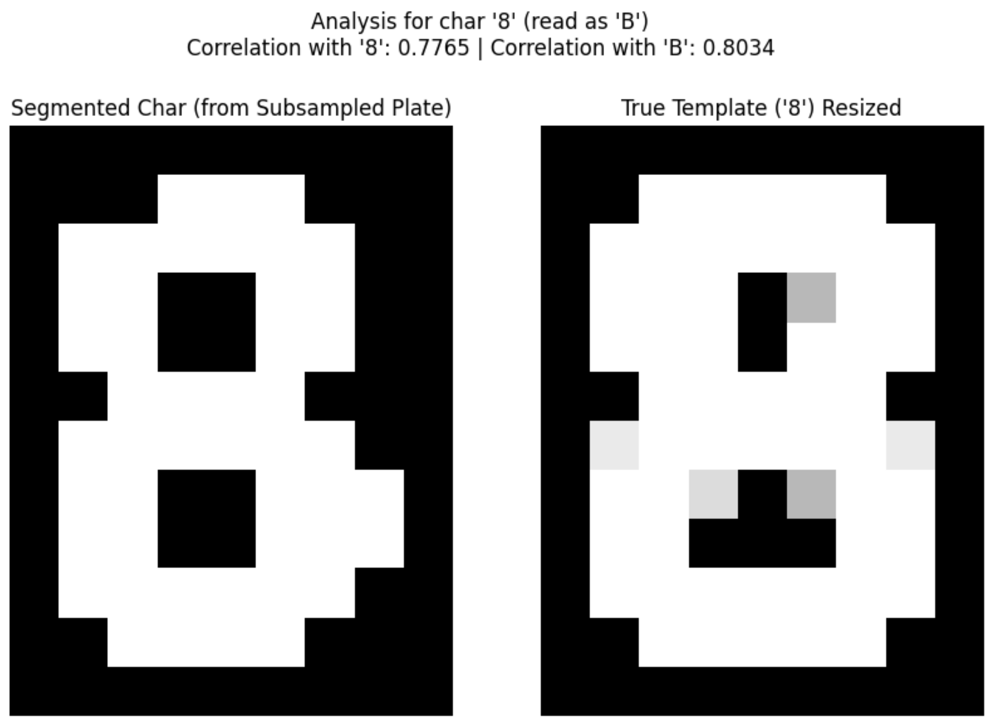
Visual Analysis

To better understand the results, the following plots were generated at the breaking point (N=9).

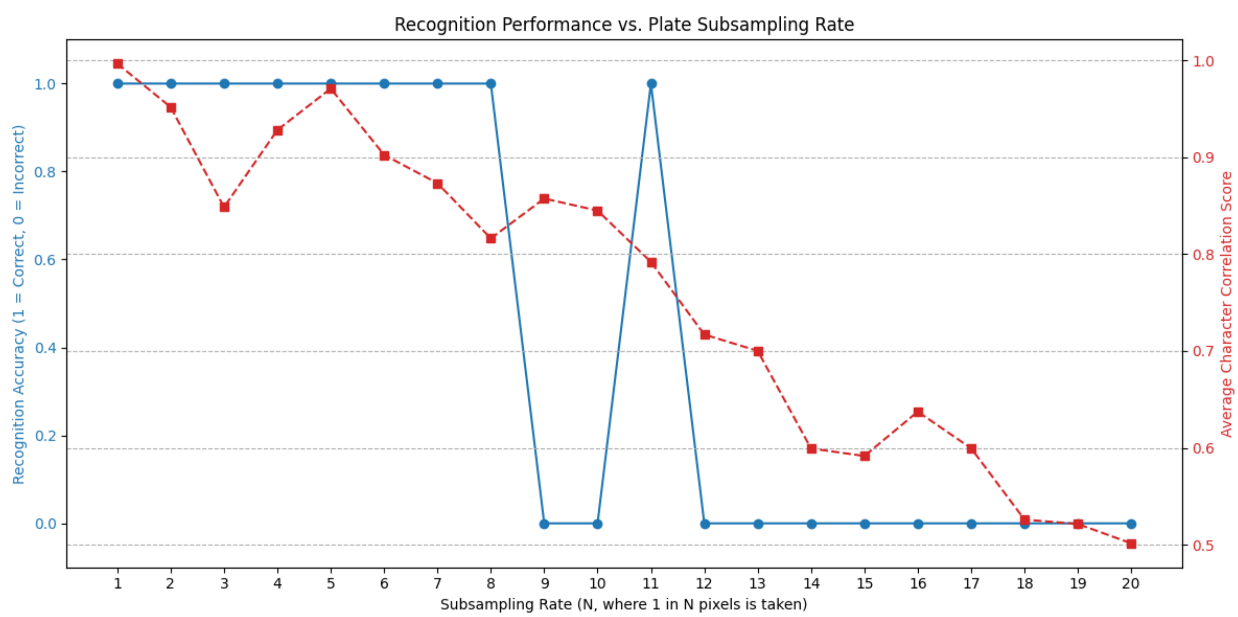
- **Plot 1: Comparison of Down-sampling Methods** This plot clearly illustrates the visual difference between the two methods. The image on the left (Subsampling) has a harsh, jagged appearance due to pixel removal, while the image on the right (Resize) appears soft and blurry due to pixel combination.



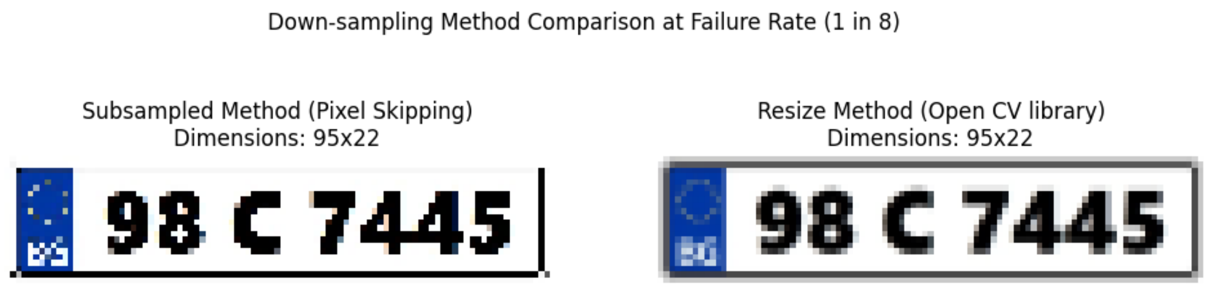
- **Plot 2: Analysis of the Failing Character** This plot shows the root cause of the recognition error. The character '8' extracted from the plate is compared with two reference templates, '8' and 'B'. As shown in the plot's title, its correlation coefficient with the incorrect template ('B') was slightly higher than with the correct one ('8'), leading to the misidentification.

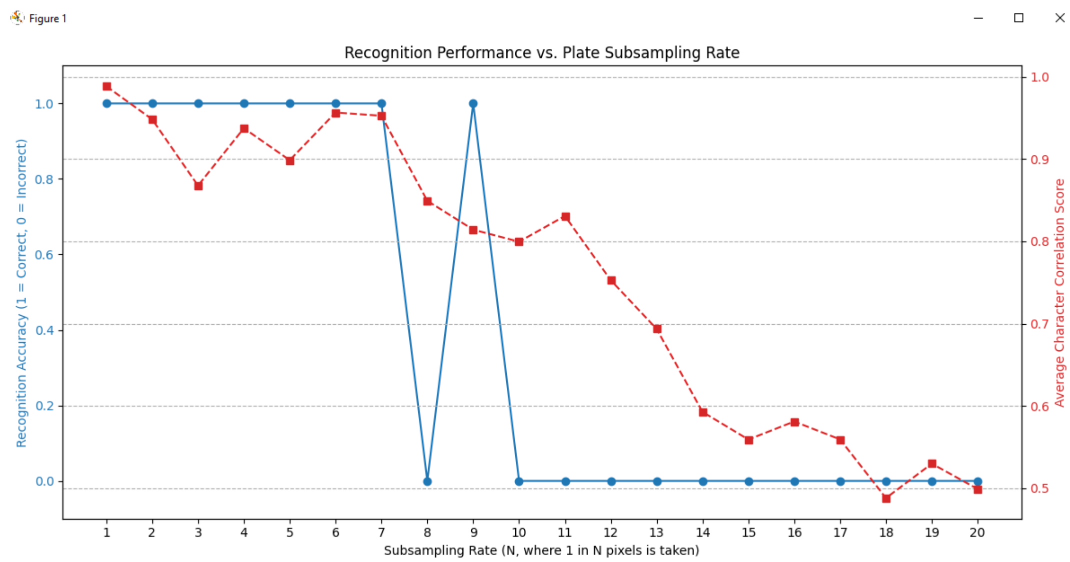


- **Plot 3: Overall Performance Graph** This comprehensive graph shows the overall trend of the system's performance. The blue line (accuracy) indicates the exact breaking point of the algorithm, while the red line (correlation) visualizes the gradual decline in the system's "confidence" as the image quality degrades.

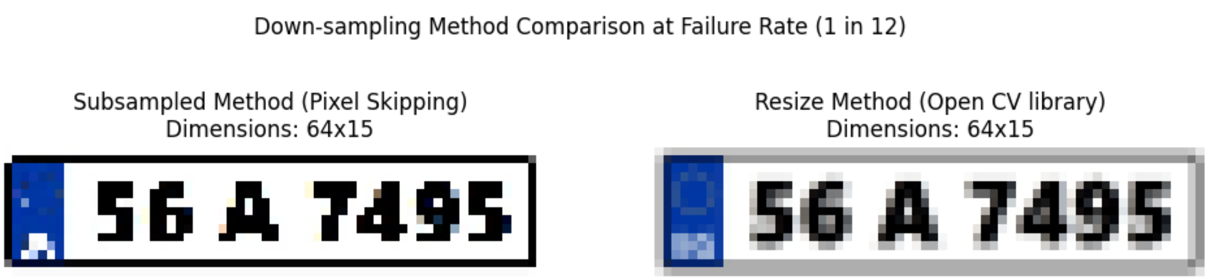


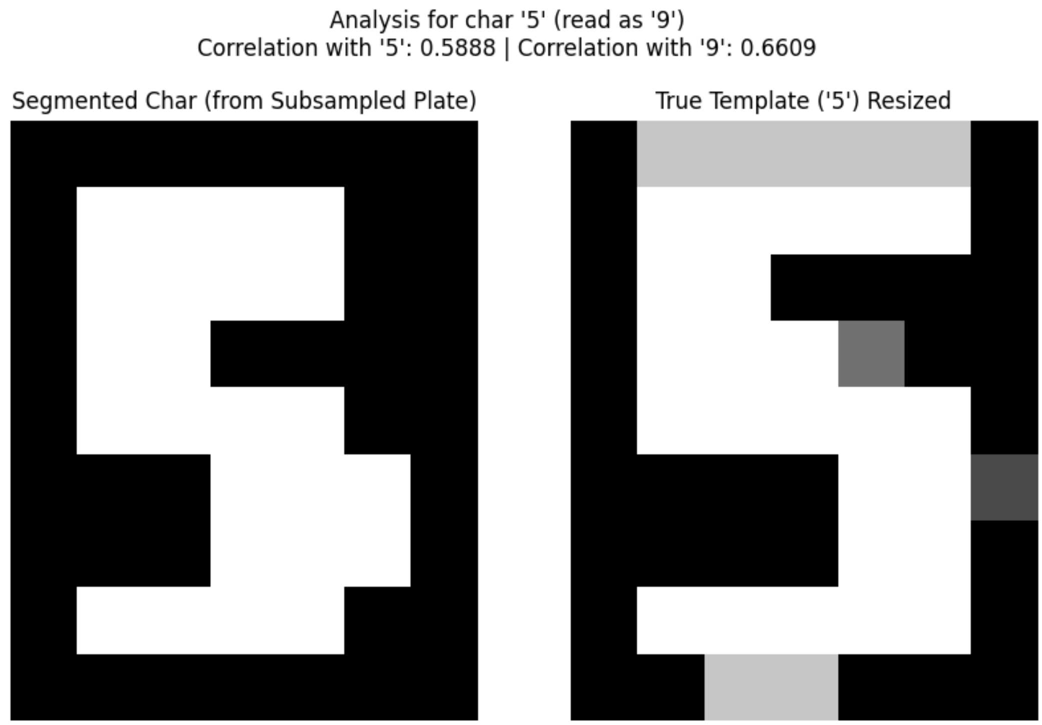
(This section output of the rest of the license plates of the cars)

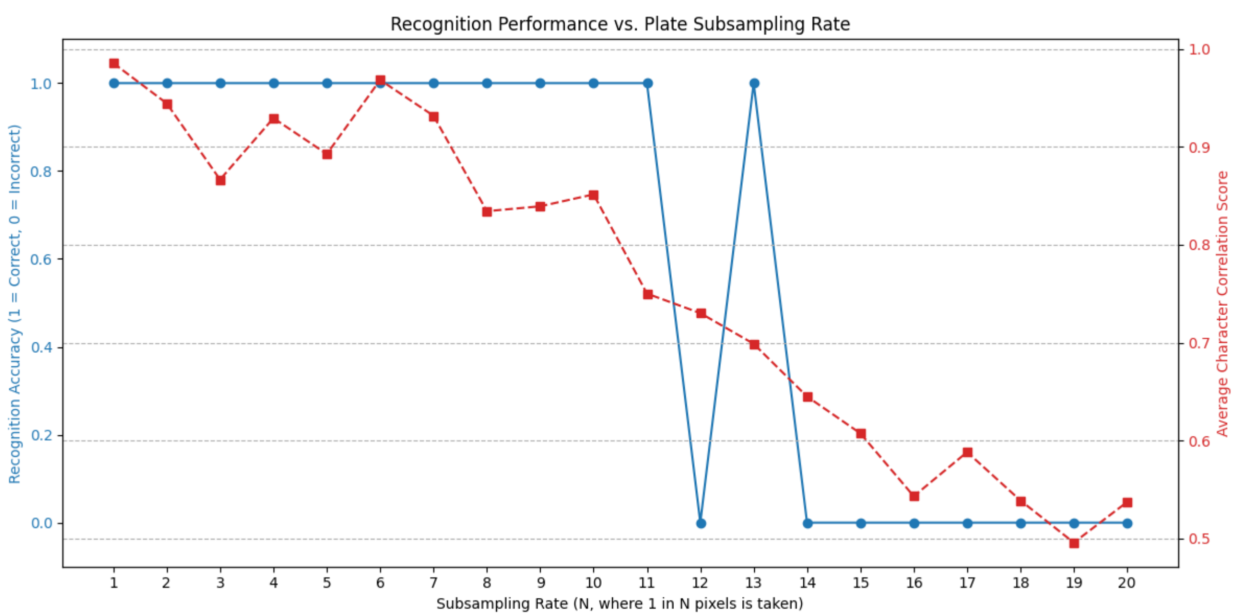




```
● Step 1: Establishing dynamic ground truth...
  -> Dynamic Ground Truth established as: '98C7445'.

  Step 2: Analyzing effects of plate subsampling...

  --- Subsampling Analysis Results ---
  Rate (1 in N)    | Recognized Plate    | Avg. Correlation    | Correct?
  ----------------------------------------------------------------------------
  1                | 98C7445             | 0.9885              | Yes
  2                | 98C7445             | 0.9483              | Yes
  3                | 98C7445             | 0.8677              | Yes
  4                | 98C7445             | 0.9375              | Yes
  5                | 98C7445             | 0.8986              | Yes
  6                | 98C7445             | 0.9565              | Yes
  7                | 98C7445             | 0.9526              | Yes
  8                | 98C75               | 0.8492              | No
  9                | 98C7445             | 0.8142              | Yes
  10               | 96C75               | 0.7996              | No
  11               | C76                 | 0.8309              | No
  12               | 98C74               | 0.7527              | No
  13               | 96076               | 0.6936              | No
  14               | 676                 | 0.5928              | No
  15               | 475                 | 0.5592              | No
  16               | A8419               | 0.5811              | No
  17               | 98C75               | 0.5594              | No
  18               | 41A                 | 0.4881              | No
  19               | AA474               | 0.5299              | No
  20               | 5AC75               | 0.4990              | No
```




Analysis for char '5' (read as '9')
Correlation with '5': 0.5888 | Correlation with '9': 0.6609
Segmented Char (from Subsampled Plate)
True Template ('5') Resized


Recognition Performance vs. Plate Subsampling Rate
Recognition Accuracy (1 = Correct, 0 = Incorrect)
Average Character Correlation Score
Subsampling Rate (N, where 1 in N pixels is taken)

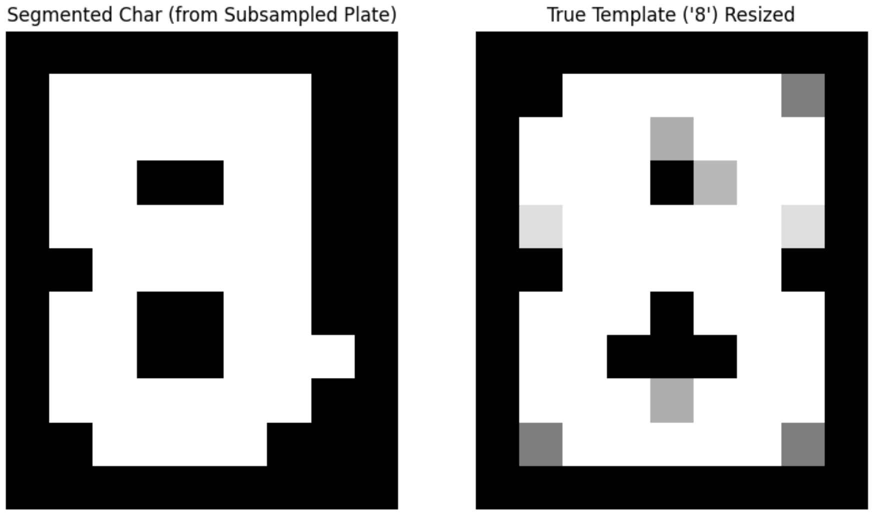
```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '56A7495'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)   | Recognized Plate     | Avg. Correlation     | Correct?
---------------------------------------------------------------------------
1               | 56A7495              | 0.9856               | Yes
2               | 56A7495              | 0.9445               | Yes
3               | 56A7495              | 0.8666               | Yes
4               | 56A7495              | 0.9294               | Yes
5               | 56A7495              | 0.8927               | Yes
6               | 56A7495              | 0.9684               | Yes
7               | 56A7495              | 0.9319               | Yes
8               | 56A7495              | 0.8344               | Yes
9               | 56A7495              | 0.8394               | Yes
10              | 56A7495              | 0.8515               | Yes
11              | 56A7495              | 0.7502               | Yes
12              | 56A7499              | 0.7302               | No
13              | 56A7495              | 0.6989               | Yes
14              | 66A7496              | 0.6450               | No
15              | 84A19                | 0.6074               | No
16              | 5847AA5              | 0.5431               | No
17              | 56A7A95              | 0.5883               | No
18              | 94A7499              | 0.5388               | No
19              | 475                  | 0.4958               | No
20              | AA87                 | 0.5373               | No
```

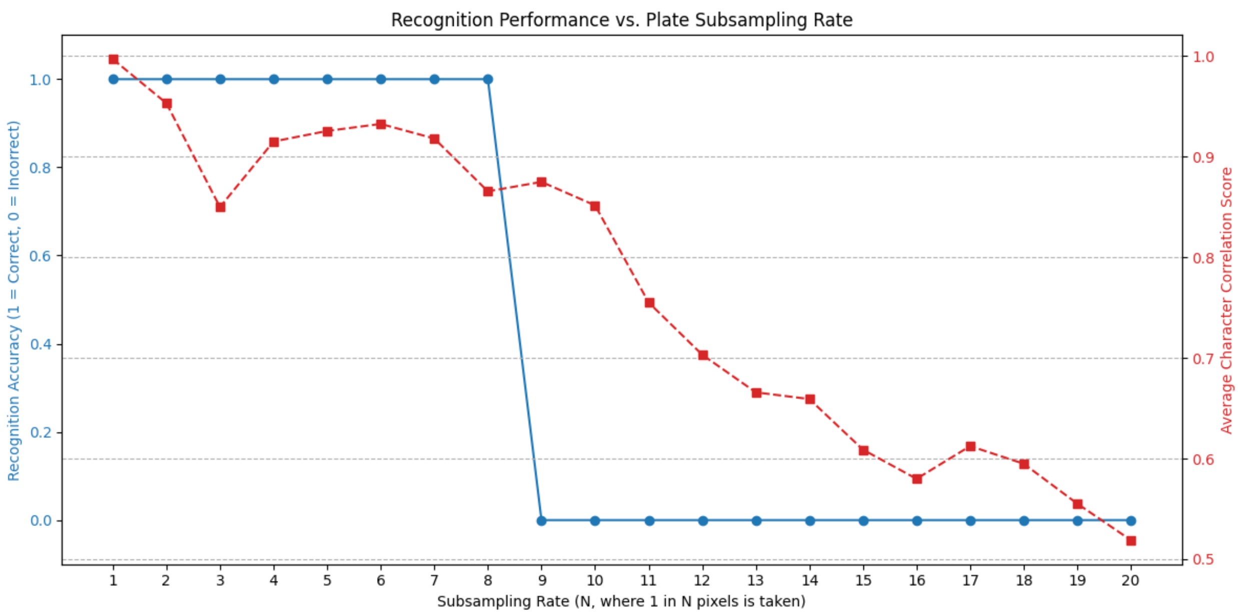
Down-sampling Method Comparison at Failure Rate (1 in 9)

Subsampled Method (Pixel Skipping)
Dimensions: 85x19

Resize Method (Open CV library)
Dimensions: 85x19

Analysis for char '8' (read as 'B')
Correlation with '8': 0.7827 | Correlation with 'B': 0.8262





```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '79B1208'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)   | Recognized Plate     | Avg. Correlation     | Correct?
---------------------------------------------------------------------------
1               | 79B1208              | 0.9970               | Yes
2               | 79B1208              | 0.9532               | Yes
3               | 79B1208              | 0.8499               | Yes
4               | 79B1208              | 0.9151               | Yes
5               | 79B1208              | 0.9256               | Yes
6               | 79B1208              | 0.9326               | Yes
7               | 79B1208              | 0.9181               | Yes
8               | 79B1208              | 0.8655               | Yes
9               | 79B120B              | 0.8750               | No
10              | 7901206              | 0.8516               | No
11              | 81208                | 0.7551               | No
12              | 7981208              | 0.7032               | No
13              | 81298                | 0.6658               | No
14              | 819                  | 0.6591               | No
15              | 81218                | 0.6088               | No
16              | 81A                  | 0.5799               | No
17              | 79848                | 0.6124               | No
18              | 837AA                | 0.5949               | No
19              | 84AAA                | 0.5552               | No
20              | 1A84                 | 0.5189               | No
```

"Realistic case"

Part I : Attempt to extract the license plate characters from realistic folder:

```
PS C:\Users\mirsh> & C:/Users/mirsh/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/mirsh/Desktop/S&S_Project/ideal.py"
Step 1: Segmenting characters from the license plate...
Could not proceed with recognition as no characters were segmented.
```

The main reason for this error is the destructive effect of blur on the character edges. Blur transforms sharp boundaries into a soft gray gradient that is unrecognizable to the algorithm.

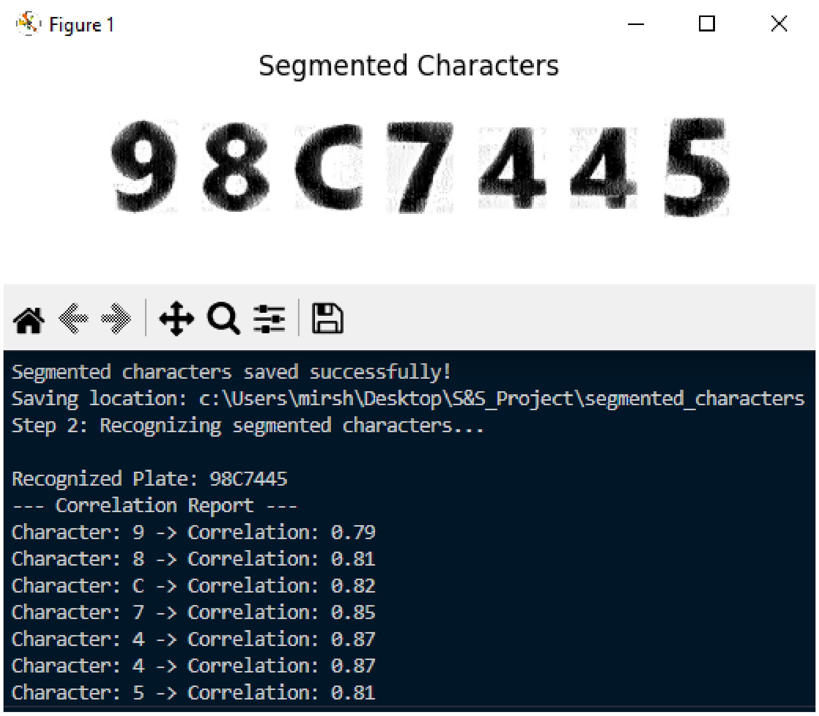
During the thresholding stage, these soft edges cause the character shapes in the black-and-white image to become fragmented. As a result, the contour detection function identifies numerous incomplete and scattered contours instead of finding the complete outlines of the characters.

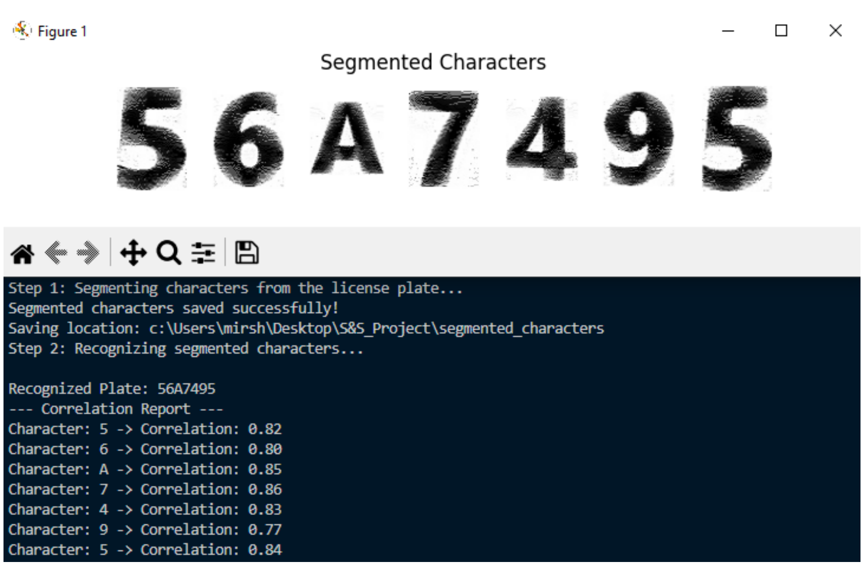
In the final stage, the algorithm filters these contours based on their dimensions. Since none of these fragments match the size of a real character, they are all rejected, and the final list remains empty.

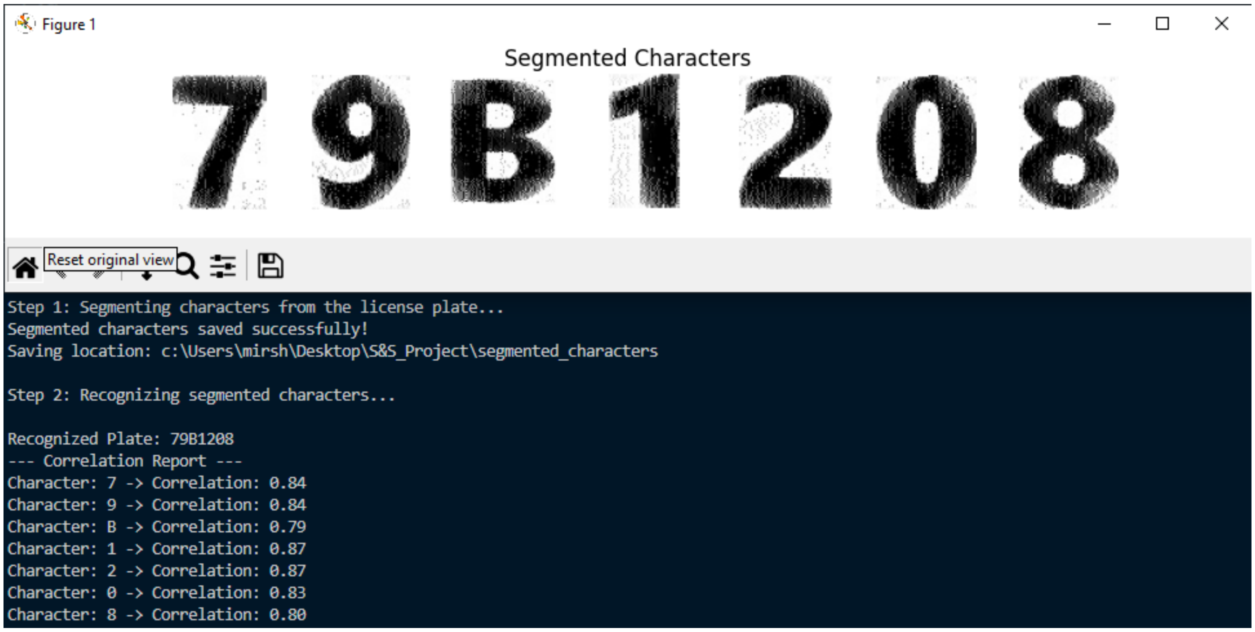
Part II & III & VI : The report for Parts 2 & 3 & 6 of the project is in *realistic.pdf* , and the corresponding code is in *realistic.ipynb*.

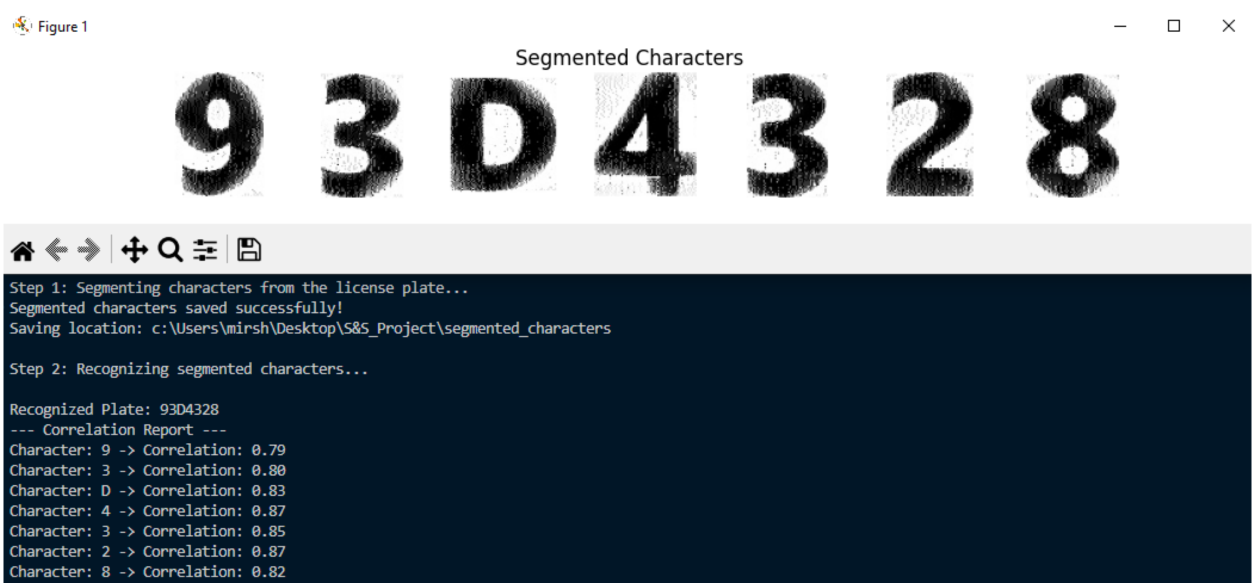
Part IV : Character Recognition on Deblurred Images

The outputs from the *realistic.ipynb* script, which are the deblurred images, are saved in the results folder. In this section, these images are fed into the *ideal.py* character recognition script. The recognition results for each deblurred license plate are presented below.









Part V : Analysis of the Robustness of the License Plate Recognition Algorithm Against Subsampling on deblurred Realistic Images (*ideal_resize_analysis.py*)

In this section, the deblurred images that were saved in the results folder will be fed into the downsampling analysis script originally written for the ideal case. As the algorithm for this script has been explained previously, only the outputs are presented below.

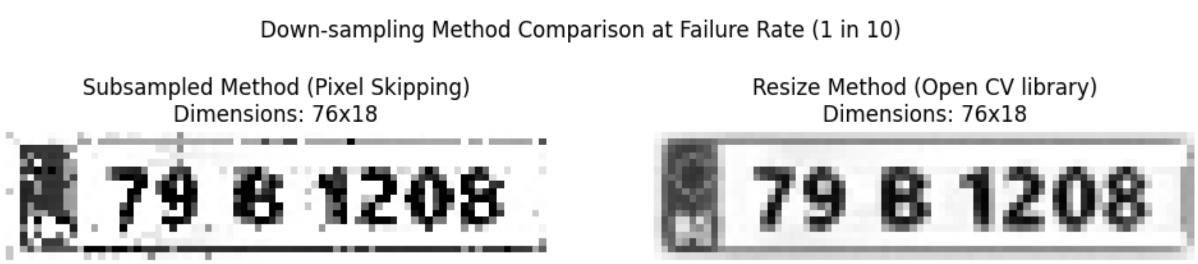
Quantitative Results

```
PS C:\Users\mirsh> & C:/Users/mirsh/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/mirsh/Desktop/S&S_Project/ideal_resize_analysis.py"
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '79B1208'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)    | Recognized Plate      | Avg. Correlation      | Correct?
---------------------------------------------------------------------------
1                | 79B1208               | 0.8317                | Yes
2                | 79B1208               | 0.8427                | Yes
3                | 79B1208               | 0.8251                | Yes
4                | 79B1208               | 0.8135                | Yes
5                | 79B1208               | 0.8320                | Yes
6                | 79B1208               | 0.8105                | Yes
7                | 79B1208               | 0.7801                | Yes
8                | 79B1208               | 0.7175                | Yes
9                | 79B1208               | 0.8211                | Yes
10               | 79012A8               | 0.7367                | No
11               | 7901208               | 0.7166                | No
12               | 1961248               | 0.6350                | No
13               | 7981108               | 0.5925                | No
14               | 84309                 | 0.5599                | No
15               | 812A8                 | 0.5440                | No
16               | 74A1AA4A              | 0.4682                | No
17               | 79848                 | 0.5368                | No
18               | 1383766               | 0.5025                | No
19               | A7246                 | 0.4913                | No
20               | 7484                  | 0.4585                | No

-> Failure point identified at 1-in-10 pixels. Generating comparison plots...

Visualizing the difference between Subsampling and Resizing at rate 1-in-10...
-> The character 'B' at index 2 was misread as '0'.
```

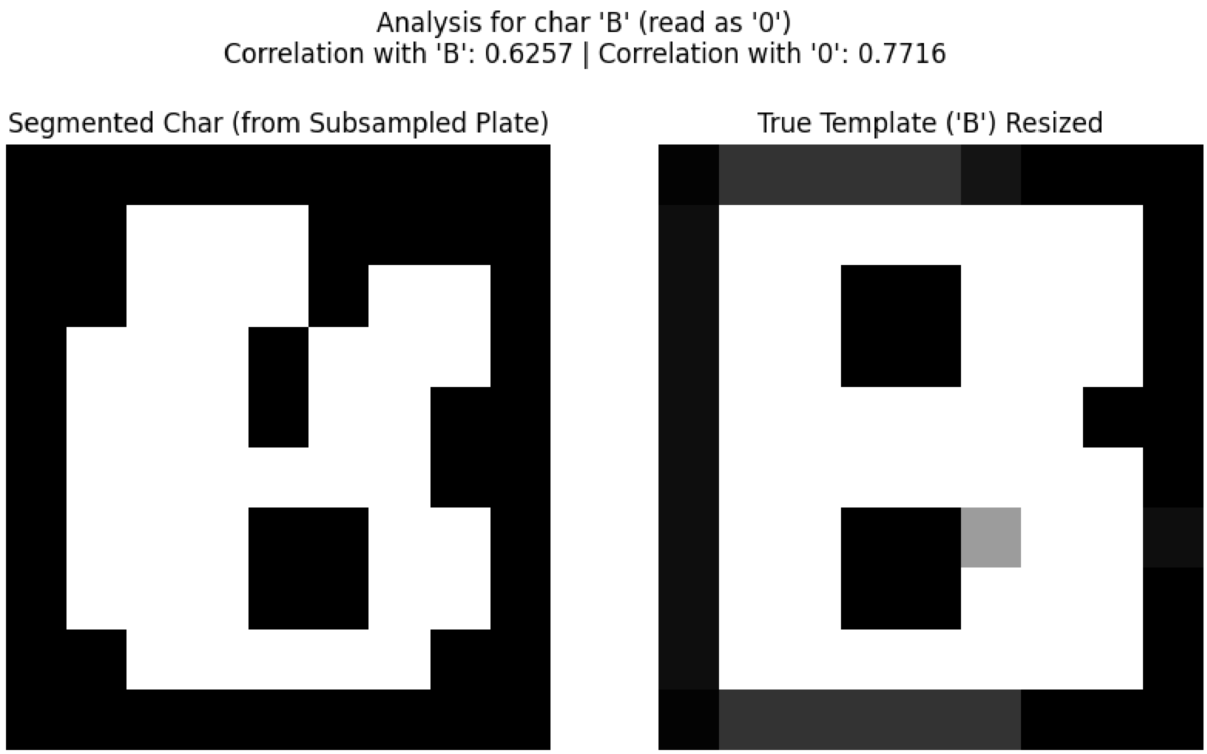
Visual Analysis

To better understand the results, the following plots were generated at the breaking point (N=10).

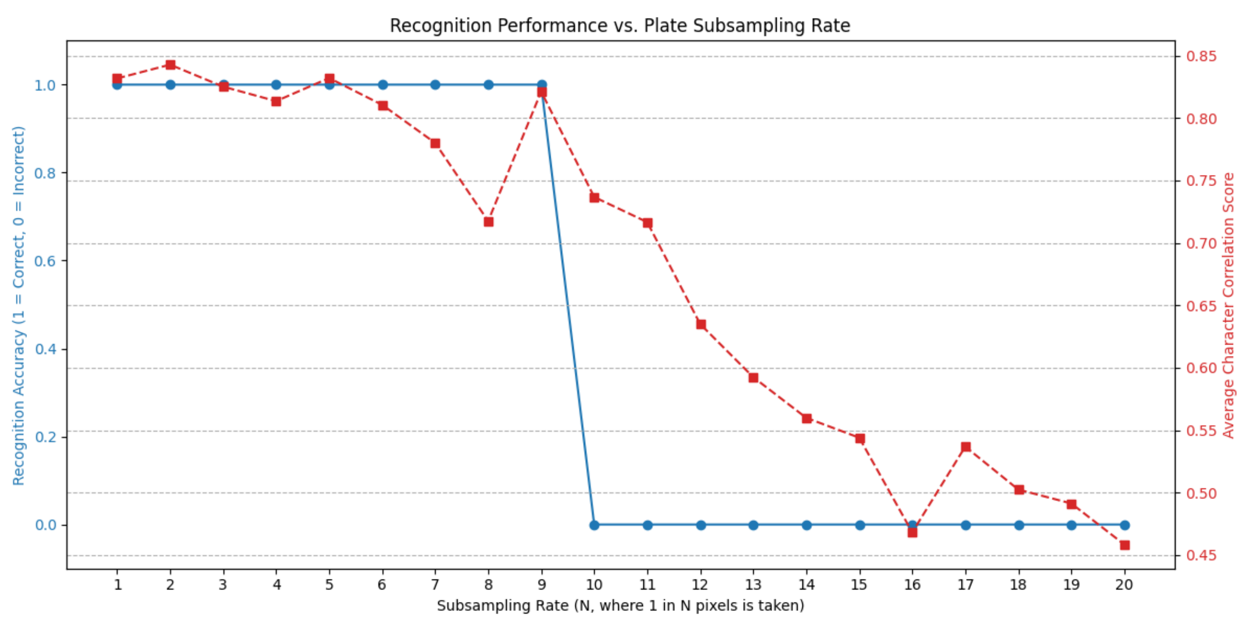
- **Plot 1: Comparison of Down-sampling Methods**



- **Plot 2: Analysis of the Failing Character**

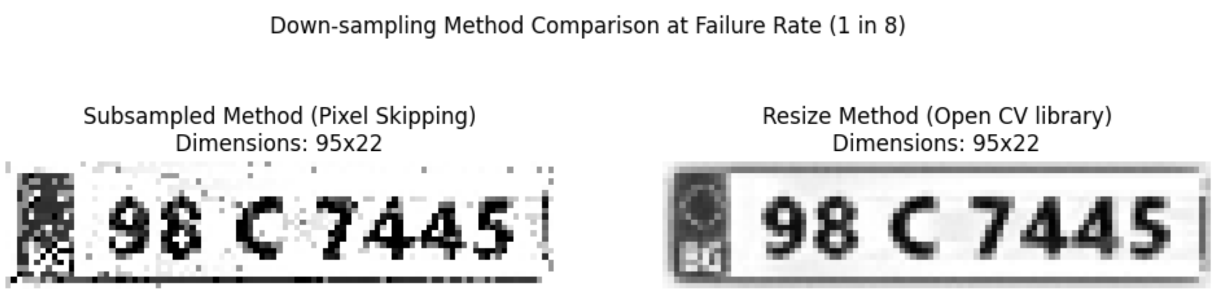


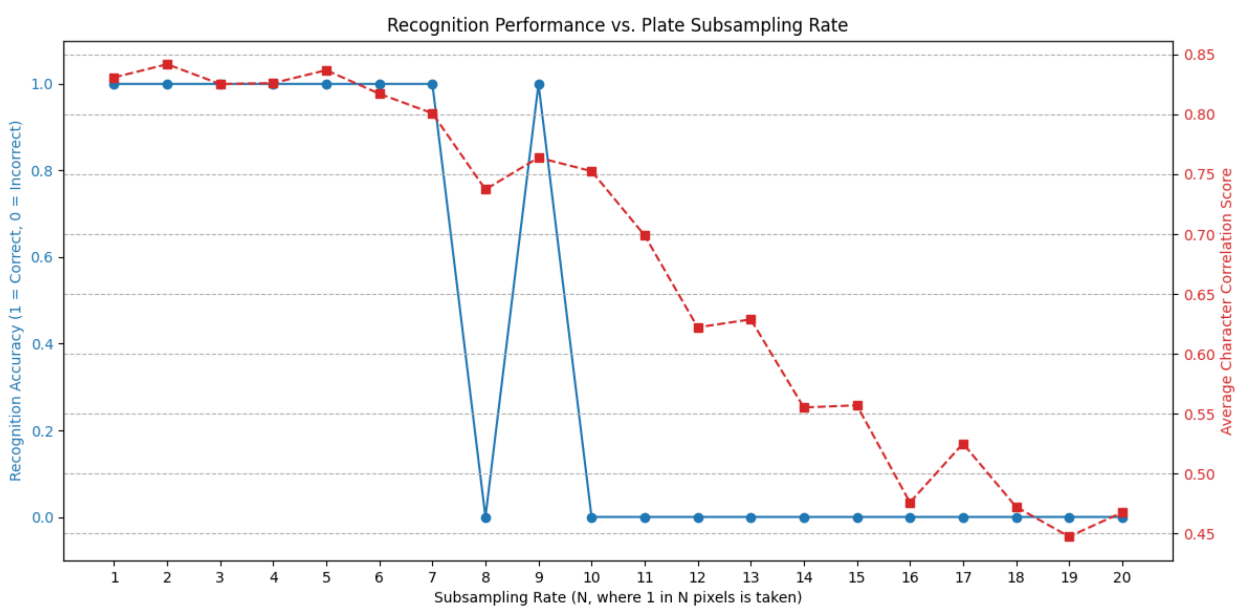
- **Plot 3: Overall Performance Graph**



(This section output of the rest of the license plates of the cars)

```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '98C7445'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)   | Recognized Plate      | Avg. Correlation      | Correct?
-----------------------------------------------------------------------------
1               | 98C7445               | 0.8309                | Yes
2               | 98C7445               | 0.8419                | Yes
3               | 98C7445               | 0.8253                | Yes
4               | 98C7445               | 0.8261                | Yes
5               | 98C7445               | 0.8369                | Yes
6               | 98C7445               | 0.8172                | Yes
7               | 98C7445               | 0.8009                | Yes
8               | 98C79                 | 0.7373                | No
9               | 98C7445               | 0.7639                | Yes
10              | 98C75                 | 0.7526                | No
11              | 58C75                 | 0.6990                | No
12              | 98C1449               | 0.6221                | No
13              | 98C76                 | 0.6288                | No
14              | 9667449               | 0.5552                | No
15              | C75                   | 0.5570                | No
16              | 5844A44               | 0.4759                | No
17              | 98C15                 | 0.5251                | No
18              | 3A41A                 | 0.4723                | No
19              | AA474                 | 0.4474                | No
20              | 3A37335               | 0.4676                | No

-> Failure point identified at 1-in-8 pixels. Generating comparison plots...
```



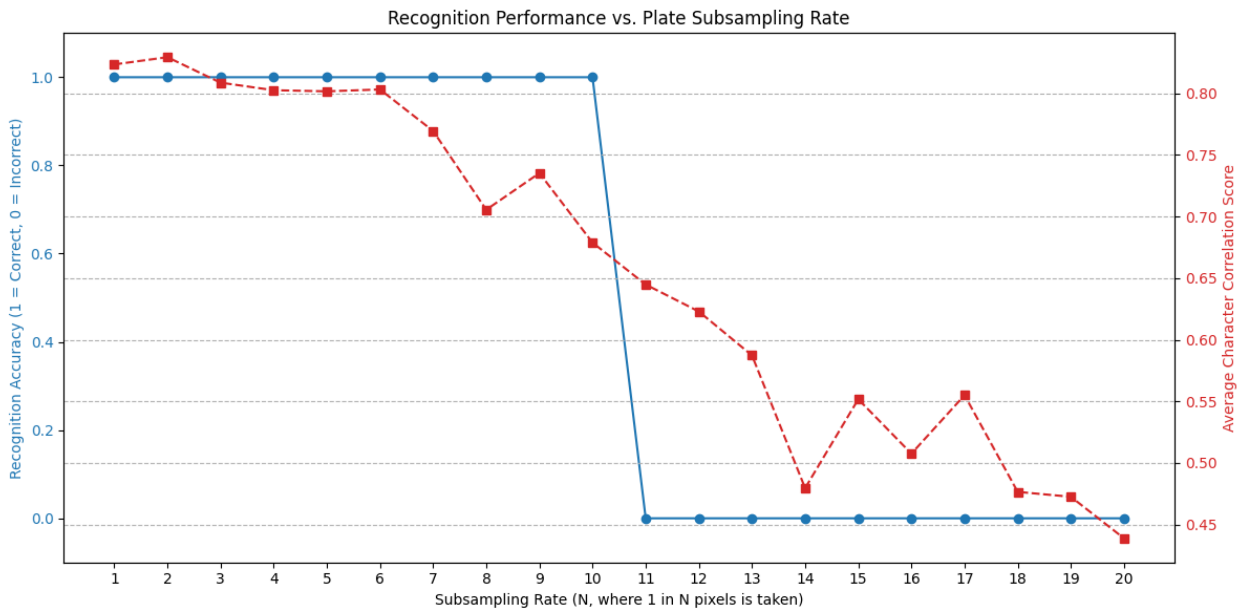


```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '56A7495'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)   | Recognized Plate     | Avg. Correlation     | Correct?
--------------------------------------------------------------------------
1               | 56A7495              | 0.8233               | Yes
2               | 56A7495              | 0.8292               | Yes
3               | 56A7495              | 0.8086               | Yes
4               | 56A7495              | 0.8024               | Yes
5               | 56A7495              | 0.8015               | Yes
6               | 56A7495              | 0.8030               | Yes
7               | 56A7495              | 0.7694               | Yes
8               | 56A7495              | 0.7054               | Yes
9               | 56A7495              | 0.7352               | Yes
10              | 56A7495              | 0.6788               | Yes
11              | 94A7A95              | 0.6448               | No
12              | 56A7465              | 0.6229               | No
13              | 56A7895              | 0.5873               | No
14              | 1642495              | 0.4796               | No
15              | 58475                | 0.5517               | No
16              | 5AAAAA               | 0.5076               | No
17              | 56A7295              | 0.5550               | No
18              | 9447A99              | 0.4765               | No
19              | 4A4784A              | 0.4726               | No
20              | 468A                 | 0.4389               | No

-> Failure point identified at 1-in-11 pixels. Generating comparison plots...
```

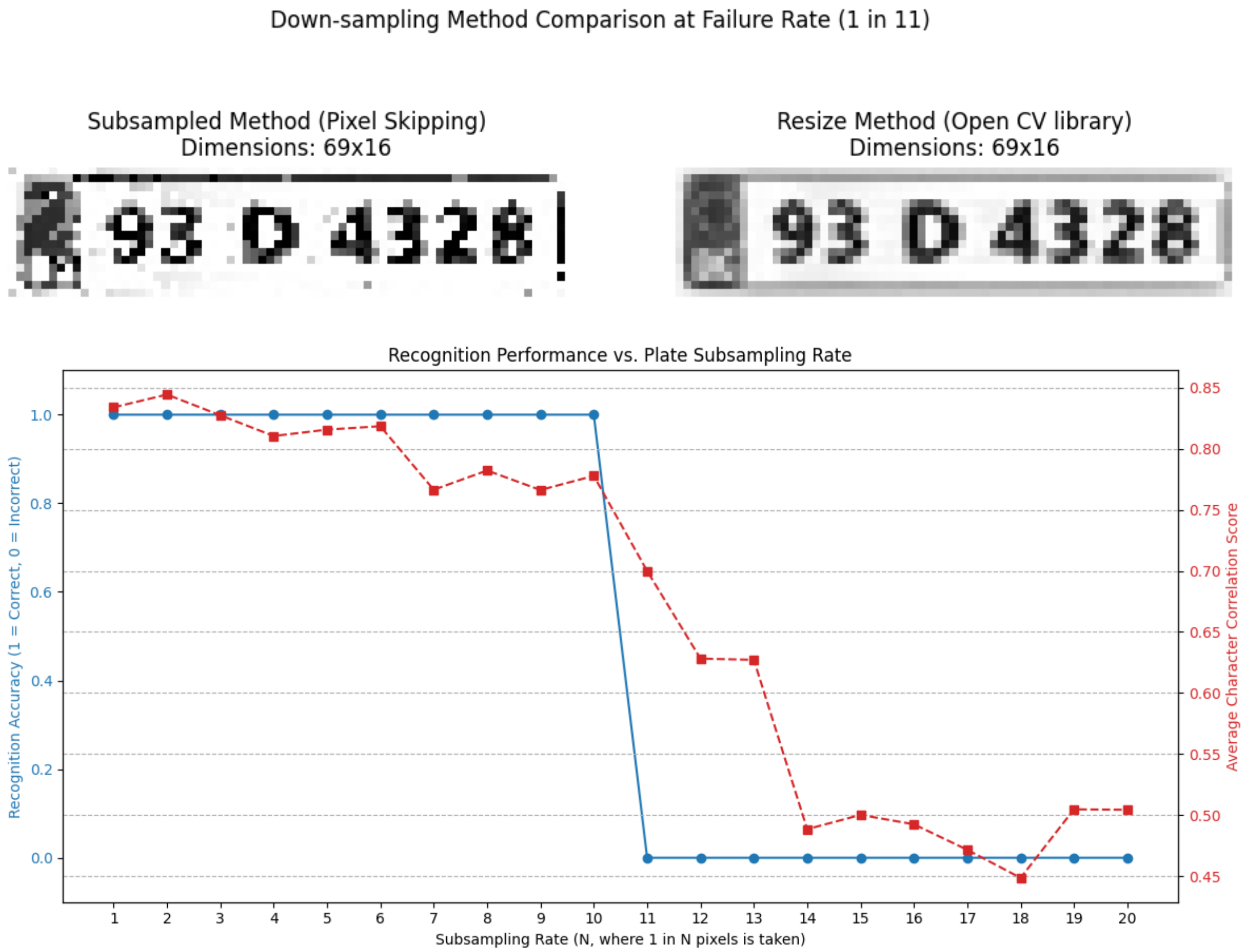
Down-sampling Method Comparison at Failure Rate (1 in 11)

Subsampled Method (Pixel Skipping)
Dimensions: 69x16

Resize Method (Open CV library)
Dimensions: 69x16



```
Step 1: Establishing dynamic ground truth...
-> Dynamic Ground Truth established as: '93D4328'.

Step 2: Analyzing effects of plate subsampling...

--- Subsampling Analysis Results ---
Rate (1 in N)    | Recognized Plate      | Avg. Correlation      | Correct?
--------------------------------------------------------------------------
1                | 93D4328               | 0.8339                | Yes
2                | 93D4328               | 0.8444                | Yes
3                | 93D4328               | 0.8274                | Yes
4                | 93D4328               | 0.8104                | Yes
5                | 93D4328               | 0.8157                | Yes
6                | 93D4328               | 0.8185                | Yes
7                | 93D4328               | 0.7664                | Yes
8                | 93D4328               | 0.7822                | Yes
9                | 93D4328               | 0.7662                | Yes
10               | 93D4328               | 0.7778                | Yes
11               | 9304328               | 0.6999                | No
12               | 9304328               | 0.6282                | No
13               | 9304328               | 0.6271                | No
14               | 9104331               | 0.4885                | No
15               | 31066                 | 0.5002                | No
16               | A70AA                 | 0.4925                | No
17               | A38A388               | 0.4716                | No
18               | 910617A               | 0.4486                | No
19               | 0812A                 | 0.5047                | No
20               | A403A                 | 0.5044                | No

-> Failure point identified at 1-in-11 pixels. Generating comparison plots...
```



Down-sampling Method Comparison at Failure Rate (1 in 11)
Subsampled Method (Pixel Skipping)
Dimensions: 69x16
93 D 4328
Resize Method (Open CV library)
Dimensions: 69x16
93 D 4328
Recognition Performance vs. Plate Subsampling Rate
Recognition Accuracy (1 = Correct, 0 = Incorrect)
Average Character Correlation Score
Subsampling Rate (N, where 1 in N pixels is taken)
0.0
0.2
0.4
0.6
0.8
1.0
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20