Image Preprocessing in Improvement of American Sign Language Classification

N.D.P. Tran(2731525), C.E. Mulder(2753754), J.C. McLuckie(2698617), J.G.C Appelman(2686547), S.D. Conte(2739767)

Vrije Universiteit Amsterdam
De Boelelaan 1105, 1081 HV Amsterdam, Netherlands

Abstract

Machine learning is crucial in advancing the accuracy and efficiency of ASL recognition, making it an indispensable tool for bridging the communication gap between deaf and hearing communities. This study aims to investigate whether the application of transformation filters can enhance the recognition accuracy of CNN models. Specifically, the study examined the impact of Gaussian blur and Canny filter as preprocessing techniques on the recognition accuracy of convolutional neural network (CNN) models for American Sign Language (ASL) recognition. The findings indicate that utilizing these filters results in decreased accuracy rather than improving it. Hence, it is apparently recommended to avoid their usage in preprocessing for ASL recognition with CNN models.

Keywords: Convolutional Neural Network · Image Recognition · American Sign Language · Image Preprocessing.

1 Introduction

For decades, the communication gap between the hearing-impaired community and the general population has posed a significant challenge. This challenge extends across various domains, such as human-computer interaction, sign language recognition, robotics, sustainable development, and equal opportunities, highlighting the importance of interdisciplinary perspectives in addressing this issue [1].

To tackle this challenge, researchers have developed a plethora of sophisticated methods for recognizing and interpreting ASL gestures. For instance, a recent study in Pakistan showcased a prototype that integrated CNNs into Google Class, enabling immediate recognition of PSL gestures [2]. Although the impact of image preprocessing on feature extraction has not received widespread attention from the scientific community, there are still notable examples in the literature that explore the efficacy of transformation filters in this context.

Gaussian blur and Canny filter, two widely-used image transformation techniques, have garnered particular interest due to the various advantages they offer in enhancing feature extraction. These approaches have been instrumental in improving the performance of image recognition systems by reducing noise and extracting meaningful features from images, which can subsequently be used to train classification models such as CNNs.

Overall, the use of these approaches for feature extraction has been shown to be effective in improving the accuracy of image recognition systems. These techniques help to reduce noise and extract meaningful features from images, which can then be used to train classification models such as CNNs. Studies as Sainath et al.'s [3] and Sun et al.'s [4] demonstrate the effectiveness of these techniques in different applications, highlighting their versatility and importance in the field of computer vision. Sainath et al. compared the performance of various feature extraction techniques, including Gaussian blur and Canny filter, for the task of hand gesture recognition using the American Sign Language (ASL). The authors found that Gaussian blur was effective in reducing noise and smoothing the image, while Canny filter helped to extract edges and contours. The combination of these techniques was found to be particularly effective in improving recognition accuracy for ASL hand gestures.

Additionally, in the discussion of the paper "Recognition of Wood Defects Based on Convolutional Neural Networks and Image Processing Techniques" [4], the authors found that Gaussian blur was effective in reducing noise and improving the quality of the images, while Canny filter was effective in extracting edges and contours. The use of these techniques for feature extraction was found to significantly improve the accuracy of the CNN in classifying wood defects. The advantages of these techniques stem from their ability to enhance the salient features in the input image, which can help to improve the accuracy of the CNN model.

Our research primarily concentrates on the influence of transformation filters in image processing models for ASL recognition, particularly in the context of convolutional neural networks (CNNs). There are two common methods for image

transformation: noise suppression and edge detection. Noise suppression aids models in maintaining invariance to marginal image changes while preserving essential details. Previous research [5] demonstrates that utilizing the Adaptive Gaussian filter minimizes the Mean Squared Error (MSE) of average models, subsequently enhancing edge extraction performance compared to non-filter models. Edge detection, on the other hand, distinguishes primary subjects from irrelevant backgrounds [6]. However, the efficiency of these transformations varies across models, depending on the original data characteristics and experimenters' central focus. Therefore, this study aims to answer the following question: "How do various filter configurations affect the accuracy of a CNN model for image classification tasks of ASL sign language?"

This question holds substantial significance as improved ASL recognition accuracy could bridge the communication gap between the hearing and deaf communities. The implementation of CNN models combined with diverse filter configurations for ASL recognition offers a promising approach to enhancing recognition accuracy. Our hypothesis posits that specific filter configurations will significantly impact the CNN model's accuracy for ASL image classification tasks. By examining the effectiveness of different filter configurations, this research aims to contribute to the identification of optimal preprocessing techniques capable of augmenting the accuracy of CNN models for ASL recognition.

2 Methodology

2.1 Approach

Firstly, the datasets were explored and analyzed to determine their characteristics. Secondly, a set of preprocessing filters were selected that are suitable for the chosen datasets. Thirdly, we constructed the reasonable sequences of filters to investigate their efficiency in the isolated and combined context. The first set exclusively consists of the baseline model without any filters applied. The second set would apply the Gaussian blur before training the model. The last set is the use of Canny filter accompanied by the closing filter to create a smooth edge, which is the input for the baseline. Then, we train such processed images in a fundamental CNN model. Our assumption is that all models have no different outcomes between the color and grayscale images, thus all models learn from the grayscale image as colour is not important for this use-case. Ultimately, statistical validation is performed on the results to answer the addressed questions.

2.2 Exploratory Data Analysis

Dataset 1(ASL Alphabet) comprises of 87000 images with 29 classes (depicted in Figure 10). Each class contains 3000 instances, therefore it can be concluded that this dataset does not contain any class imbalance

Dataset 2(Sign Language (ENG alphabet)) comprises of 24000 images with 24 classes (depicted in Figure 11). Each class contains 1000 images, and similarly to dataset 1, there are no signs of class imbalance.

Therefore, all datasets have a uniform distribution of instances over classes.

2.3 Basic Data Manipulation

As mentioned in 1.2, we would process colourless images, so we first transform all images in the datasets to grayscale. Since each dataset was taken under different conditions, they have different sizes. Next, the images were resized to the same size (24 pixels by 24 pixels), which means the image has 24 pixels for each dimension.
Additionally, both datasets do not have completely similar classes; therefore, we only include classes that are present in both datasets. Our final dataset contains 24 classes which are all letters of the English alphabet, excluding J and Z (since these two letters do not appear in Dataset 2).

2.4 Gaussian Blur

The Gaussian filter is introduced to simulate the standard normal distribution on a kernel. The middle of the matrix corresponds to the mean of the standard normal distribution (the highest possibility to achieve). The other values are symmetric around the middle value and they grow bigger as they move to the middle (presented in Figure 1).

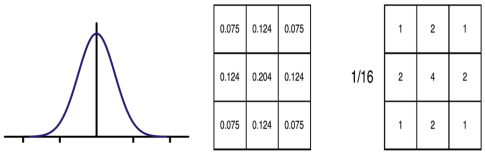

Figure 1: The visualization of normal distribution and the corresponding Gaussian kernel

The center of the matrix (4) is the highest possible value and the values surrounding it are lower the further they move from the center (respectively 2 and 1). The main aim of the Guassian filter is to remove the detail and noise from the image.
When we apply the Gaussian Blur on images, the resolution would decrease (depicted in Figure 2), but it promotes the invariance of the model with respect to minimal changes of images.

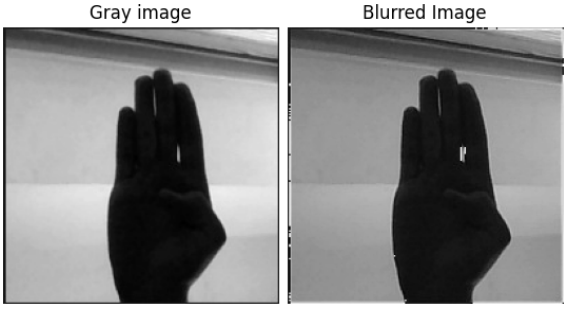

Figure 2: The effect of Gaussian Blur in our dataset

2.5 Canny filter and closing

The canny filter is used to determine the different edges in the images. It has 5 steps and it makes use of the procedure of the Gaussian filter [7, 8]. The five steps of the canny filter are:

1. Apply the gaussian filter to smooth the noise and details
2. Find magnitude and direction of the gradient intensity of the image
3. Apply gradient magnitude threshold
4. apply double threshold to find potential edges
5. Track image by hysteresis

For the first step we have to apply the Gaussian kernel as illustrated in the previous section. In step two we want to find the gradient intensity of the image. There are three different gradients possible: vertical (Gv), horizontal(Gh) and diagonal (G). The Canny filter makes use of the edge detection operator of Robberts, Prewitt and Sobel to find the different gradients(respectively Gv,Gx and G). After the edge detection operator method we can calculate the gradient of the edge with the use of Pythagoras formula. So the gradient magnitude can be calculated by [7, 8]:

$$G = Gx2 + Gy2 \tag{1}$$

Now we found the gradient magnitude we want to determine the gradient direction.The direction of the gradient is an conversion from Cartesian coordinates to polar coordinates and can be calculated with:

$$(i, j) = arctan(Gy/Gx) \tag{2}$$

These directions are divided in 8 equal subsections(presented in Table 1.

0-45	45-90
90-135	135-180
180-225	225-270
270-315	315-360

Table 1: 8 subsections of polar coordinates

You can use the gradient of this pixel to determine where this pixel is pointing to. The next step is to compare the pixel magnitude of the gradient with the magnitude of the gradient which it is pointing to. If the gradient of the pixel is lower than the pixel it is pointing then the result is that this could never be an edge point and the strength of the pixel is set to 0. If the gradient magnitude is higher than the corresponding pixel in the direction then this pixel might be an edge point. In step four we are going to decide whether a pixel is an edge point. We set a lower and upper threshold to determine whether a magnitude of the gradient from the remaining pixels satisfies this condition. So for instance, if the gradient is higher than 1 it is stated that this is an edge point. If the gradient is lower than 0.5 it is stated that it is not an edge point. For the values in between, we need to look at eight neighborhood pixels and determine whether they are higher than the upper threshold or not. If one of the gradients of the neighborhood pixels is higher than the threshold then the pixel is stated as an edge point.

The major advantage of the canny edge filter is that it is a good detection technique that has low probability of failing to mark true edge points and it has low probability of falsely marking non edge points [7]. Besides that it has good localization and low spurious response [7]. Our research aim is to classify the different hand language signs to the alphabet with different filters. The use of the Canny filter can achieve that because the contours, and therefore, the edges of the hand are vital in determining the different classes. This is why we stated that the canny filter is important in the CNN model.

Erosion is the process that the location of pixels assigned a value of one in the structuring element is similar such positions in the input image, such areas would be eroded and set to the original value. In contrast, dilation would increase such overlapped areas between structuring element(SE) and input image(IM). Closing composes of dilation and erosion [9]. More accurately, it is the procedure that the dilation in the inpute image(IM) is followed by the erosion in the structuring element(SE). The closing operator is defined as below [10]:

$$Closing = IM \oplus SE \ominus SE \tag{3}$$

Consequently, the elements in background which are smaller than the structuring elements would be eliminated, and the output image is enhanced considerably by noise suppression and emphasizing the primary features in the input picture.

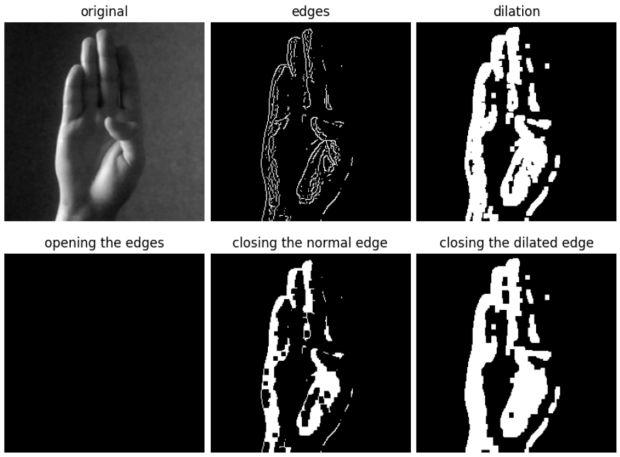

Figure 3: The effect of Canny filter and Closing process

In Figure 3, we applied Canny filter in the colourless image (original) to extract the edge of the hand (edges). Then, from these edges, we dilated them over one iteration to gain thicker lines (dilation). Ultimately, we input both edges over the closing procedure and received 2 smoother edges (closing the normal edge and closing the dilated edge, respectively). By observation, the sequence of Canny filter accompanied with the dilation and closing processes was established for further experiment since it created a smoother and nicer foreground for the output image.

2.6 Baseline Convolution Neural Network model

Convolution Neural Networks (CNNs) are a commonly used deep neural network that is mostly responsible for image processing tasks [11]. The idea was inspired by the visual cells of living creatures that can detect, analyse, and process the light of environment. Recently, CNN has laid a foundation for computer vision industry in processing one of the most sophisticated tasks of humans.

CNN has countless variations of components; thus, we would explain it under the scope of our baseline model. There are 3 important elements that are not indistinguishable from other neural networks are Convolution layer, Pooling layer, and Flatten layer(depicted in Figure 12). The role of the convolution layer is learning feature representation of the given input by sliding the filter over the image. The weights in the filter are learned through the training phase and shared between the neighbours inside the image. The feature map (output of the convolution layer) then feed into the (Max) pooling layer to extract the most contributory pixels inside one neighbour of the image. It means that for each neighbour of 9 pixels, the pooling layer would return the pixels that has the largest value. Thus, it lowers the burden of expensive computational power by reducing the image size. However, the combination of convolution layer and pooling layer drops the resolution of the image, people normally increase the number of channels to preserve as most most significant features. Finally, the reduced layers would be flattened into a vector to be trained through a fully connected neural network (ANN). It is optional to have a batch normalisation layer to make the training more stable and faster.

Inside the fully connected neural, we have placed one Dropout layer to set some arbitrary inputs to zero at frequency rate as 0.2 for each stage in the training step. Thus, it prevents the high chance of overfitting in the ANN. Following that, two hidden layers with a ReLU activation function are arranged to decrease the non-linearity of the feature. A softmax function would then be applied to the output, which then returns a distribution over 24 classes.

2.7 Experimental Set-up

Model 1: Baseline model would be trained with stochastic gradient descent through 10 epochs, the batch size is set to 32 and the learning rate is set as 0.001 to ensure the network would not overshoot the local minimum. Computation of loss would ultilize the cross-entropy between the actual values and the predicted values. This model is used in the following models too, with the difference being the baseline model has no filters added to it, and therefore learns explicitly from the grayscale images. The hyperparameters are kept constant throughout models, as the aim of the paper is to study the effects of the applied filters, not the CNN itself.

Model 2: Baseline model accompanied with the Gaussian Blur. The kernel size is adjusted as 3x3, and the standard deviation for both dimensions are remained as default.

Model 3: Sequential edge filters of Canny filter followed by the dilation and closing processes before input into the baseline. In Canny filter, both high threshold and low threshold are set equally to 60. Both the dilation and closing processes are implemented only one iteration with a 3x3 kernel is a unit matrix.

Since all instances in one class of a dataset have relatively similar features from background to foreground, it leads to the concern of overfitting even in the testing set. Hence, we set the dataset 1 for training and the dataset 2 for testing to guarantee the validity of the outcomes. In the training phase, 80% of instances of dataset 1 are used for training and the rest is used for the fine-tuning process (validation). In testing phase, we resample 20% instances of the dataset 2 randomly and test 100 rounds to observe the average accuracy and its mean squared error (MSE). Consequently, we would be able to make a conclusion about the significant difference between models. Besides, we also plot the confusion matrix of the model to reason its future outcomes.

3 Results

In this section, we present the results of a series of experiments conducted on the three aforementioned models to investigate the impact of image transformations on ASL image processing. Our analysis focuses on the differences in accuracy between the baseline model and the models incorporating Gaussian blur or edge detection filters. We configured the Convolutional Neural Network as described in Section 2.6, utilizing a training set derived from Dataset 1 with an 80:20 split for training and validation, respectively. We examined the effect of various filter sizes and combinations on the model's performance by calculating the F1 score using the formula 4:

$$F_1 = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Filter	precision	recall	F1-score
Baseline	0.12	0.12	0.12
Canny	0.07	0.05	0.05
Gauss	0.12	0.12	0.12

Table 2: The Precision, Recall, F1 score of Baseline model, Gaussian Blur with Baseline model, and Baseline model with the sequence of Edge Emphasis filters

Due to the poor precision and recall of all three models(presented in Table 2, it leads to the majority of instances having F1 scores equal to zero. This was primarily caused by more than half of the classes exhibiting a precision or recall of zero(presented in Figures 13,14, and 15).

For each of the Gaussian, Canny, and Baseline filters, we computed the mean accuracy (μ), accuracy standard deviation (σ), loss, and confidence intervals (CI). The Canny filter exhibited the lowest mean percentage of filtered pixels (5.12%) compared to the Baseline (12.39%) and Gaussian (11.87%) filters, indicating that it is less aggressive in filtering pixels. The Canny

filter also had the lowest standard deviation (0.018) compared to the Baseline filter (0.027) and the Gaussian filter (0.026), suggesting that the Canny filter is more consistent in its pixel filtering performance than the other two models. Regarding loss values, the Gaussian filter yielded the lowest value (19.54) compared to the Baseline model (26.91) and the Canny filter (163.97). This indicates that the Gaussian filter is more effective in minimizing the loss associated with filtering pixels.

In summary, the results suggest that the Gaussian filter performs best in terms of minimizing pixel filtering loss, while the Canny filter demonstrates less aggressive pixel filtering and greater consistency in its filtering process. These findings provide insights into the impact of different filter configurations on the accuracy of CNN models for ASL image processing tasks.

Filter	μ%	std%	loss%
Baseline	12.39%	0.027%	26.91%
Canny	5.12%	0.018%	163.97%
Gauss	11.87%	0.026%	19.54%

Table 3: The mean, standard deviation, and loss of three models

We proceeded to calculate the confidence intervals (CI) for the mean percentage of pixels filtered for the Canny and Gaussian filtering techniques, as displayed in Table 4. This table presents the CI for each filter, the degree of overlap between the CIs, and the conclusions drawn based on this overlap.

Filter	95% CI%	Overlap	Conclusion
Baseline	12,39 +/- 1,7E-04	NO	p<0,0001
Canny	5,12 +/- 1,12E-04	NO	p<0,0001
Gauss	11,87+/- 1,63E-04	NO	p<0,0001

Table 4: The Confidence Intervals of three models

The values in Table 3 indicate that the mean percentage of pixels filtered differs for each filter, as demonstrated by the non-overlapping confidence intervals. The relatively small standard errors suggest that the estimates of the means are likely to be precise. The conclusion column reveals that the p-value for each filter is less than 0.0001, signifying that the differences between the mean percentage of pixels filtered by each filter are statistically significant. Overall, these results imply that there are statistically significant differences in the mean percentage of pixels filtered by each filter.

To conclude our statistical analysis, we computed precision, recall, and F1 scores (Figures 13,14, and 15), which are widely used metrics in machine learning for evaluating the performance of classification models, particularly for imbalanced datasets. In this case, the Gaussian filter exhibits better values in comparison to the Canny filter. When comparing the Gaussian filter with the baseline model, similar values are reported, albeit with a marginally lower F1 score for the baseline model. As a result, both Gaussian Blur and the sequence of Edge detection filters made no significant contributions in addressing ASL recognition problem.

4 Discussion

In this study, we aimed to investigate the effects of different image filters, namely Gaussian and Canny filters, compared to no filters, on the accuracy of an ASL image processing model. Our findings reveal that there are statistically significant differences ($p<0.0001$) in the performance of these filters. The baseline model yielded the highest accuracy (12.39% CI +/- 1.7E-04), followed by the Gaussian filter (11.87% +/- 1.63E-04) and the edge detection filters (5.12% +/- 1.12E-04). Despite the observed differences, the low accuracy rates across all models raise concerns regarding their practical applications in facilitating effective communication.

The nature of ASL could have contributed to the low accuracy rates, as some ASL letters have similar visual features, leading to potential misclassification. For instance, the letters P and G (displayed in Figure 7) bear a striking resemblance, which might affect the model's ability to differentiate between them (the situation has highest number of False instances in Figure 4). The confusion matrices obtained from the experiments also support this notion, as we observed a high rate of misclassification among visually similar letters such as E, N, T, and S.

Another finding is that the Canny filter demonstrated perfect precision for the letter Y(depicted in Figure 6, suggesting the potential for developing specialized models capable of accurately predicting specific letters. Combining such models in an ensemble could potentially improve overall accuracy and reduce errors in predicting the entire alphabet.

Despite the low F1 scores observed for all models, the results offer valuable insights into the model's performance in differentiating ASL letters. The F1 score intervals for baseline, Gaussian, and Canny filters were 0; 0.30, 0; 0.31, and 0; 0.13, respectively, indicating that none of the models could provide adequate classification performance for any letter. In fact, many letters were not predicted at all, resulting in an F1 score of 0.

The models were heavily biased towards predicting specific letters. For example, the baseline and models predominantly predicted the letter K and G (dislayed in Figure 4 and similarly the model with Gaussian filter project all letters as G like it momentum, while the Canny filter model favored the letter N. This suggests that blurring an image (Gaussian filter) retains classification properties similar to the baseline model, whereas edge-detection (Canny filter) shifts the model's predictions towards different letters. Overall, the baseline is the less biased predictor when compared to the models with the application of transformation. The evidence for this statement is laid under the false predictions are not concentrating on solely one letter, its range is broader than that of models with filters. An assumption is possibly reasoned is that the filters had eliminated some crucial elements of the image, then causes more confusion between ensemble hand gestures for the neural network (baseline model). Additionaly, the datasets inherently contain severely

less variation from the subjects to the background elements, which would apparently impose a necessary requirement for further research about ASL data exploration.

5 Conclusion

In conclusion, this study investigated the impact of various image filters, specifically baseline, Gaussian, and Canny filters, on the accuracy of a Convolutional Neural Network (CNN) model for ASL image classification tasks. The experimental results revealed statistically significant differences in the performance of the three models. However, the low accuracy rates across all models raise concerns regarding their practical applications in facilitating effective communication.

The analysis of the confusion matrices and F1 scores provided valuable insights into the models' performances in differentiating ASL letters. For instance, we observed a high rate of misclassification among visually similar letters and heavy bias towards predicting specific letters depending on the filter utilized. Interestingly, the Canny filter demonstrated perfect precision for the letter Y, suggesting the potential for developing specialized models capable of accurately predicting specific letters. Combining such models in an ensemble could potentially improve overall accuracy and reduce errors in predicting the entire alphabet.

Although the study offers valuable contributions to the understanding of the impact of different image filters on the accuracy of CNN models for ASL recognition, further research is warranted to explore other advanced preprocessing techniques and machine learning algorithms. Incorporating a Markov chain, Bayes rule, and k-order Markov chains as hyperparameters may enhance the performance and accuracy of ASL gesture recognition systems by focusing on the intended words rather than isolated letters. By optimizing filter configurations and incorporating advanced techniques, this research area has the potential to significantly improve ASL recognition accuracy, leading to more effective communication using ASL recognition models and opening up new avenues for bridging the communication gap between the hearing and deaf communities. Additionally, exploring other deep learning architectures, such as recurrent neural networks (RNN) or transformers, could provide further insights into the most effective methods for ASL image classification tasks.

Overall, this study serves as a stepping stone for future research aiming to optimize and enhance the performance of ASL gesture recognition systems. It is crucial to continue exploring and refining preprocessing techniques, machine learning algorithms, and deep learning architectures to develop more accurate and reliable ASL recognition models that can effectively facilitate communication between the hearing and deaf communities.

5.1 Future Research

While our results did not meet the initial expectations, they provided valuable insights into the challenges faced by ASL recognition models and the potential for improving performance by developing specialized models for specific letters. Future research could explore the development of such specialized models and investigate ensemble techniques to create a more accurate and reliable ASL recognition system. Additionally, further investigation into the choice of filters, hyperparameter optimization, and other preprocessing techniques could help enhance the performance of ASL image processing models.

In particular, one potential direction for future research involves incorporating a Markov chain into the models, which would enable the calculation of the probability of a specific letter appearing given the presence of preceding letters. By employing the Bayes rule and the Markov assumption, it becomes possible to compute P(L|D), where L represents the letter and D denotes the data. This approach filters out improbable letter combinations that are not typically found in words. For instance, after observing "KL," the probability of encountering "L" is zero, as "KLL" is not a likely word.

To implement this approach, the classification process should be based on the words that the user aims to predict, alongside the Markov chain. Initially, the user must define a distribution π for the words they intend to form. Subsequently, the CNN model can be applied to predict a specific letter. The Markov chain's role is to estimate the most probable outcome for the various letters provided by the CNN. For example, if S, E, and N exhibit similarities in ASL, the Markov chain predicts P(S|D), P(E|D), and P(N|D), where S, E, and N represent the different letters, and D refers to the data.

By applying the Bayes rule in a k-order Markov chain (with k as a hyperparameter), the network can then determine the most likely outcome. Consequently, the classification process shifts its focus from individual letters to entire words that the user wants to express, ultimately enhancing the performance and accuracy of the ASL gesture recognition system. This approach, which incorporates a Markov chain and Bayes rule, has the potential to significantly improve the classification of ASL gestures, particularly in cases where the gestures are visually similar. By focusing on the intended words rather than isolated letters, this method could lead to more effective communication using ASL recognition models and open up new avenues for research in the field of machine learning and computer vision.

6 Appendix

A Confusion Matrices

Model 1: Baseline model

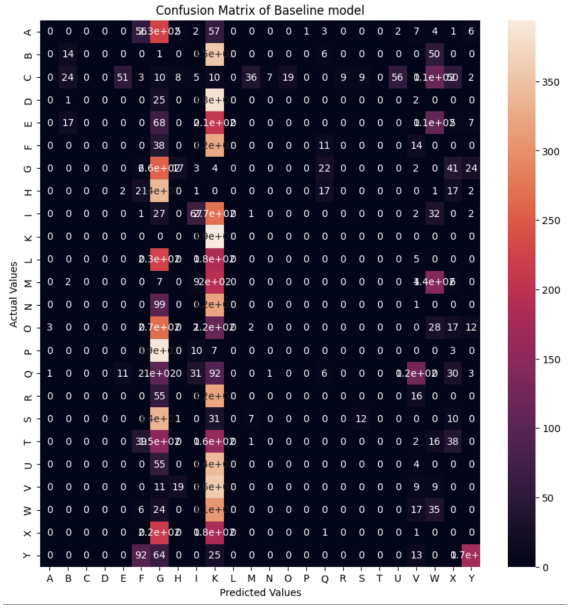

Figure 4: The confusion matrix of baseline model

Model 2: Gaussian Blur and Baseline model

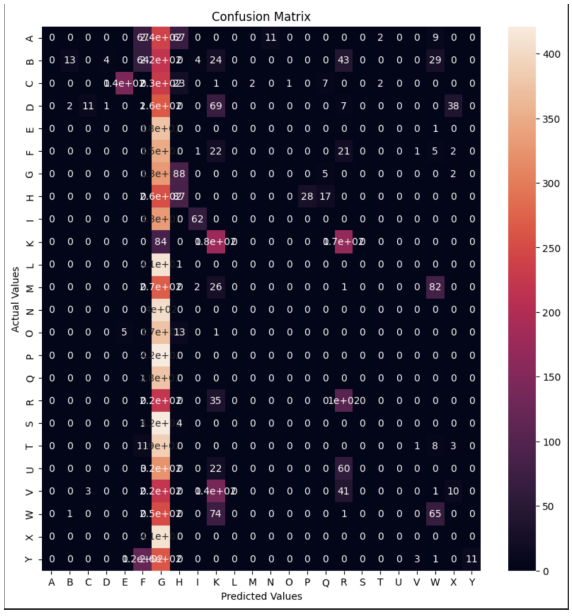

Figure 5: The confusion matrix of Gaussian Blur and Baseline model

Model 3: Sequential Edge filters and Baseline model

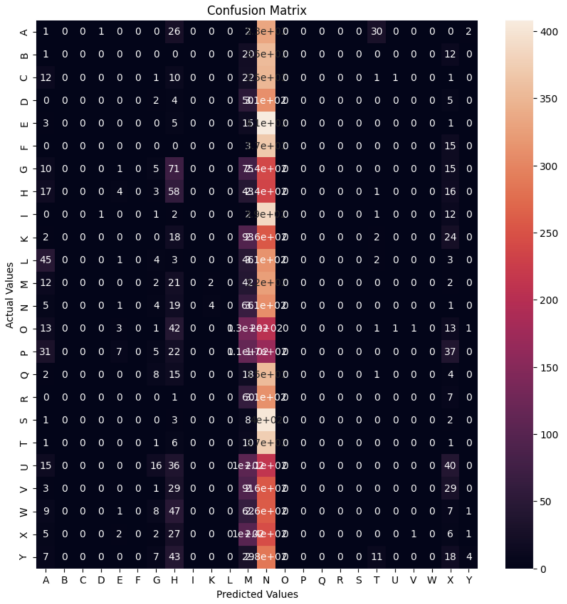

Figure 6: The confusion matrix of Sequential Edge filters and Baseline model

B Filter Comparisons

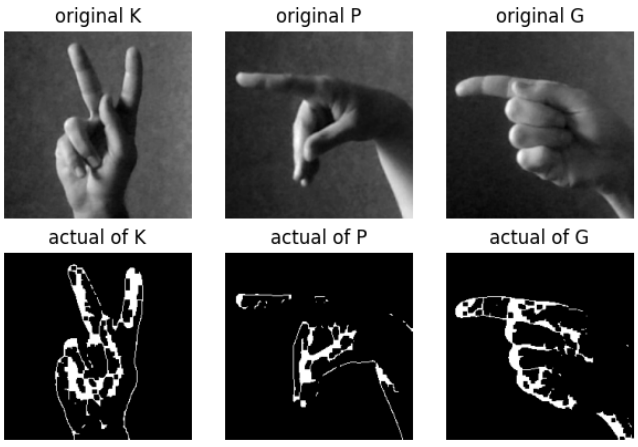

Figure 7: The significantly mistaken pairs of baseline model

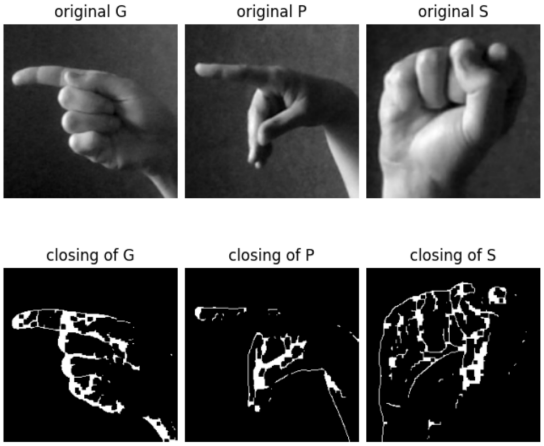

Figure 8: The significantly mistaken pairs of Gaussian Blur and Baseline model

	precision	recall	f1-score	support
A	0.21	0.06	0.09	398
B	0.54	0.03	0.06	389
C	0.24	0.12	0.16	413
D	0.11	0.04	0.06	378
E	0.07	0.03	0.04	393
F	0.00	0.00	0.00	397
G	0.04	0.15	0.06	362
H	0.00	0.00	0.00	413
I	0.59	0.21	0.31	374
K	0.12	0.93	0.21	393
L	0.00	0.00	0.00	399
M	0.00	0.00	0.00	410
N	0.00	0.00	0.00	410
O	0.00	0.00	0.00	408
P	0.00	0.00	0.00	388
Q	0.24	0.36	0.29	416
R	0.03	0.01	0.02	407
S	0.00	0.00	0.00	415
T	0.17	0.02	0.03	404
U	0.00	0.00	0.00	417
V	0.00	0.00	0.00	403
W	0.11	0.46	0.17	431
X	0.11	0.10	0.11	373
Y	0.33	0.24	0.28	409
accuracy			0.12	9600
macro avg	0.12	0.12	0.08	9600
weighted avg	0.12	0.12	0.08	9600

Figure 14: The classification report of Gaussian Blur with Baseline model

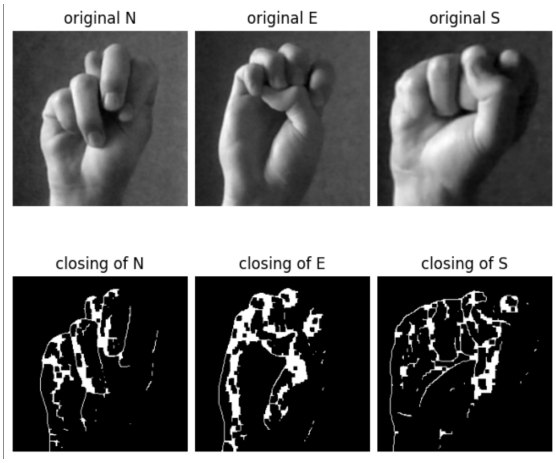

Figure 9: The significantly mistaken pairs of Sequential Edge filters and Baseline model

	precision	recall	f1-score	support
A	0.04	0.01	0.02	370
B	0.00	0.00	0.00	391
C	0.00	0.00	0.00	387
D	0.00	0.00	0.00	388
E	0.00	0.00	0.00	404
F	0.00	0.00	0.00	390
G	0.17	0.00	0.00	403
H	0.00	0.00	0.00	382
I	0.00	0.00	0.00	413
K	0.00	0.00	0.00	412
L	0.00	0.00	0.00	424
M	0.00	0.00	0.00	419
N	0.04	0.81	0.08	371
O	0.00	0.00	0.00	409
P	0.00	0.00	0.00	414
Q	0.33	0.00	0.00	402
R	0.00	0.00	0.00	387
S	0.00	0.00	0.00	369
T	0.00	0.00	0.00	413
U	0.00	0.00	0.00	406
V	0.00	0.00	0.00	399
W	0.00	0.00	0.00	412
X	0.08	0.40	0.13	424
Y	1.00	0.00	0.00	411
accuracy			0.05	9600
macro avg	0.07	0.05	0.01	9600
weighted avg	0.07	0.05	0.01	9600

Figure 15: The classification report of Canny filter with Baseline model

	precision	recall	f1-score	support
A	0.30	0.18	0.23	405
B	0.20	0.63	0.30	415
C	0.44	0.03	0.06	416
D	1.00	0.00	0.01	392
E	0.00	0.00	0.00	410
F	0.05	0.24	0.08	403
G	0.09	0.40	0.15	412
H	0.01	0.00	0.00	378
I	0.28	0.07	0.11	405
K	0.16	0.84	0.27	359
L	0.00	0.00	0.00	402
M	0.00	0.00	0.00	401
N	0.00	0.00	0.00	404
O	0.02	0.01	0.01	391
P	0.00	0.00	0.00	403
Q	0.00	0.00	0.00	389
R	0.00	0.00	0.00	376
S	0.00	0.00	0.00	399
T	0.00	0.00	0.00	404
U	0.00	0.00	0.00	381
V	0.00	0.00	0.00	412
W	0.13	0.47	0.21	402
X	0.15	0.07	0.09	417
Y	0.00	0.00	0.00	424
accuracy			0.12	9600
macro avg	0.12	0.12	0.06	9600
weighted avg	0.12	0.12	0.06	9600

Figure 13: The classification report of Baseline model

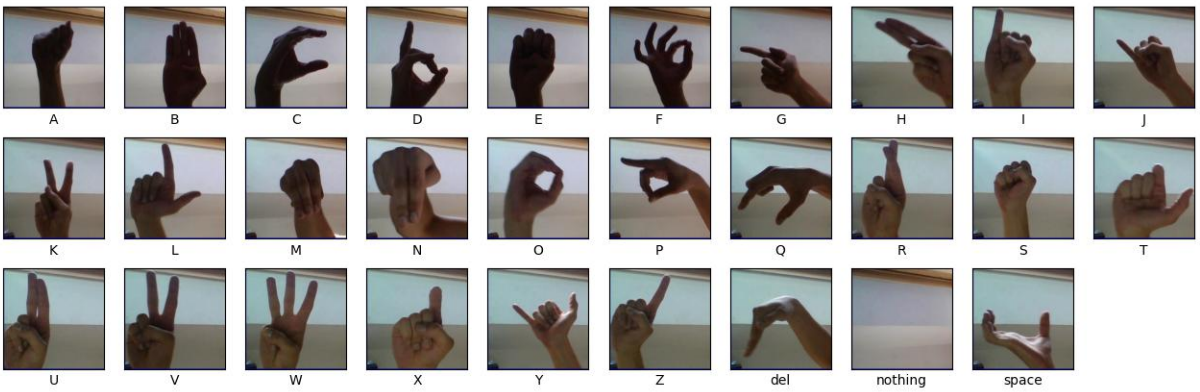

Figure 10: The instance in each class of dataset 1

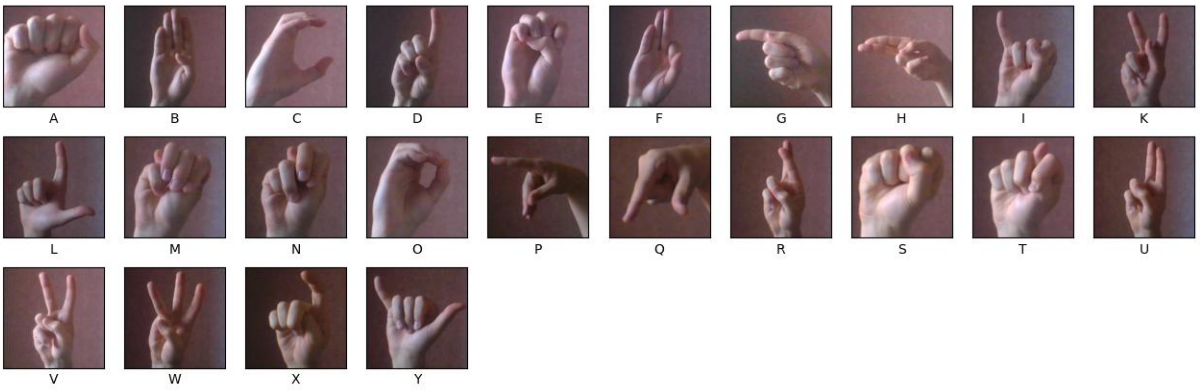

Figure 11: The instance in each class of dataset 2

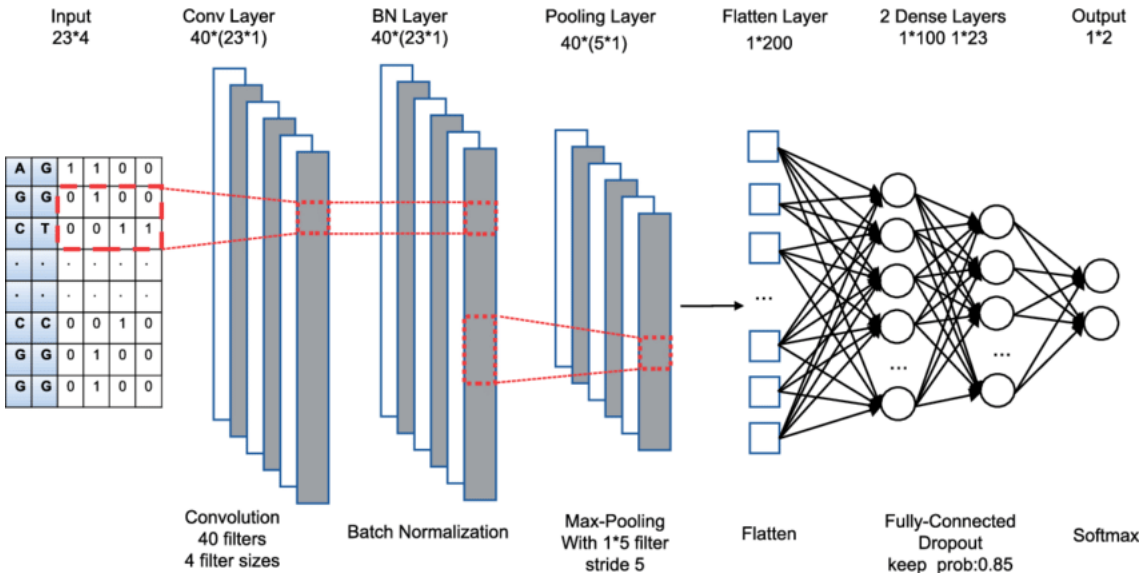

Figure 12: The architecture of our baseline model

References

[1] G De Clerck and PV Paul. Sign language, equal opportunities, and sustainable development, 2016.

[2] Maria Naseem, S Sarafraz, Ali Abbas, and Ali Haider. Developing a prototype to translate pakistan sign language into text and speech while using convolutional neural networking. *Journal of Education and Practice*, 10(15), 2019.

[3] Keerthi Sainath and P Radhakrishna Srinivasa Pai. A comparative study of feature extraction techniques for hand gesture recognition. In *International Conference on Intelligent Computing and Applications*, pages 219–226. Springer, 2017.

[4] Zhengjun Sun, Yu Zou, Wenjie Zhang, Xiaoyan Wang, Xia Liu, and Hui Lu. Recognition of wood defects based on convolutional neural networks and image processing techniques. *Sensors*, 18(7):2206, 2018.

[5] Guang Deng and LW Cahill. An adaptive gaussian filter for noise reduction and edge detection. In *1993 IEEE conference record nuclear science symposium and medical imaging conference*, pages 1615–1619. IEEE, 1993.

[6] Hasan Turhan. Comparing color edge detection techniques. 11 2012.

[7] Ruiyuan Liu and Jian Mao. Research on improved canny edge detection algorithm. In *MATEC Web of Conferences*, volume 232, page 03053. EDP Sciences, 2018.

[8] Peter Meer and Bogdan Georgescu. Edge detection with embedded confidence. *IEEE Transactions on pattern analysis and machine intelligence*, 23(12):1351–1365, 2001.

[9] Khairul Anuar Mat Said, Asral Bahari Jambek, and Nasri Sulaiman. A study of image processing using morphological opening and closing processes. *International Journal of Control Theory and Applications*, 9(31):15–21, 2016.

[10] Nannan Li, Lixin Jia, and Panpan Zhang. Detection and volume estimation of bubbles in blood circuit of hemodialysis by morphological image processing. In *2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 228–231. IEEE, 2015.

[11] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377, 2018.