Single Hand Gesture Recognition using CNN Algorithm,KNN Algorithm , Logistic Regression Algorithm and SVM Algorithm

Member 1:Kong Zi Yin
Faculty:FIST
University:MMU
Email:1201200654@student.mmu.edu.my

Member 2:Lee Wei Lun
Faculty:FIST
University:MMU
Email:1201100854@student.mmu.edu.my

Member 3:Lim Victor
Faculty:FIST
University:MMU
Email:1201100990@student.mmu.edu.my

Member 4:Tey Mei Qun
Faculty:FIST
University:MMU
Email:1201200152@student.mmu.edu.my

Abstract— **The motivation behind this project is to build a supervised learning algorithms for ASL recognition based on images. The hand sign language is important for people with hearing and discourse disabilities as that's the only way they can contact and communicate with others. We have proposed 4 methodology inside this project which are Convolutional Neural Network(CNN), K-Nearest Neighbor(KNN), Support Vector Machine(SVM) and Logistic Regression. Results obtained during this experiment have been clearly defined in the experiment section. Inside this project, Preprocessing techniques, such as noise reduction, segmentation, and feature extraction, are applied to enhance the quality of the images and extract relevant information. Based on 4 methodology, highest accuracy obtained is 0.87. Loss curve used to evaluate the with validation data. The project showcases the successful development and implementation of a machine learning algorithm for ASL recognition based on image data.**

Keywords—Hand Recognition System, Image Classification, Machine Learning, KNN, SVM

I. Introduction

Hand gesture recognition is a crucial field with a broad range of applications that extend to various domains, such as virtual reality, gaming, prosthetic hand and robot control, and communication for deaf and mute individuals. It plays a vital role in translating hand movements, specifically sign language, into words and symbols that can be understood by the general population. The ability to accurately predict hand gestures opens up numerous possibilities for enhanced human-computer interaction and accessibility.

The applications of hand gesture recognition are vast. In virtual reality and gaming, it enables users to interact with the virtual environment naturally, using hand gestures instead of traditional controllers. For prosthetic hand and robot control, accurate hand gesture recognition allows for intuitive control of these devices, mimicking natural hand movements and improving their functionality. Furthermore, hand gesture recognition can aid in communication for individuals who are deaf and mute, by translating their sign language into spoken or written words, facilitating effective communication with others.

In this context, we will present a few methods for predicting hand gestures. These methods leverage machine learning and statistical procedure techniques to analyse and interpret the intricate movements of the hand. By capturing and processing video or image data, these methods aim to accurately classify and recognize different hand gestures.

The development of accurate hand gesture recognition methods is a multidisciplinary effort, combining techniques from computer vision, machine learning, and pattern recognition. These methods typically involve extracting meaningful features from hand images and training machine learning models to classify different gestures. Deep learning algorithms, such as convolutional neural networks (CNNs) , k-nearest neighbors algorithm(KNN),and support vector machines (SVM) as well as Logistic Regression have shown promising results in this field.

By advancing hand gesture recognition techniques, we can enhance human-computer interaction, improve accessibility, and create innovative applications that benefit a wide range of users. The research and development in this field continue to push the boundaries, striving for more accurate, robust, and efficient hand gesture recognition systems.

II. Objective

In line with the objectives of TML2221, the main aim is to equip individuals with the skills to apply machine learning techniques to address real-world challenges and contribute to social welfare. Machine learning algorithms have already demonstrated their effectiveness across a wide spectrum of applications, including disease diagnosis, speech recognition, data mining, music composition, image processing, forecasting, robot control, credit approval, classification, pattern recognition, game strategy planning, compression, and many others.

The primary objective of this project is to provide participants with hands-on experience in designing a machine learning pipeline to solve practical problems. By leveraging the knowledge and tools acquired throughout the course, participants will gain valuable insights into the application of machine learning in real-life scenarios. This practical approach will enhance their ability to tackle real-world tasks and contribute to societal advancements.

III. Related Work

We collected and analyzed keywords for gestures or algorithms that we explored in our experiments across 12 papers. For non-neural network techniques, they can be divided into isolated sign language recognition and continuous sign language classification. We discovered an additional piece of information, which is a method called Hidden Markov Models (HMM) is a very powerful statistical modeling tool for speech recognition, handwriting recognition has been extensively studied in previous studies, including its modifications. HSV is good at background detection and removal, morphological manipulation and applying masks, gesture segmentation and image resizing. For neural networks techniques, Tolentino proposes using CNN (Convolutional Neural Network) to create a recognition system that can recognize numbers and additional static instead of just alphabets like other sign recognition systems. We also discover that there are other deep learning models such as Alexnet, Inception V3, Resnet50 and VGG16 that can also be used in sign recognition while maintaining a high accuracy. However, hardware resources and environment factors still serve as a limitation for both of these methods.

Besides, in order to improve the performance of SVM, Local Binary Pattern(LBP)features benefit SVM by providing a texture-rich representation, rotation invariance, dimensionality reduction, and robustness to noise. LBP captures local texture details, which helps SVM discriminate between classes effectively. The rotation invariance of LBP ensures consistent performance despite changes in object or texture orientation. LBP's conversion to decimal numbers and use of uniform patterns reduce dimensionality, preventing overfitting and speeding up processing. Additionally, LBP's ability to handle noise makes it a robust feature for SVM, improving classification accuracy in the presence of image variations.

The Speeded Up Robust Features (SURF) algorithm enhances the performance of SVM in various ways. SURF extracts robust and distinctive local features from images, ensuring invariance to scale, rotation, and affine transformations. These features enable SVM to achieve more accurate classification by capturing important image characteristics. Additionally, SURF provides efficient algorithms for feature matching, such as the Fast Hessian algorithm, which enables fast and reliable matching of extracted features. This combination of robust feature extraction and efficient matching contributes to the overall effectiveness of SVM in tasks such as object recognition, image classification, and image retrieval.

IV. Method

A. Method 1:CNN

Convolutional Neural Network is the most popular neural network model being used for image classification problems that contain one input layer, one or more hidden layers and one output layer. Each of the nodes connects to one another and has an associated weight and threshold. Moreover, Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. How does Convolutional Neural Network work?

CNN contains three main types of layers.
1. Convolutional Neural Network
2. Pooling Layer (Hidden Layer)
3. Fully-connected (FC) layer

The first layer after input layer of convolutional network will be convolutional layer, while this layer can be followed by additional convolutional layers or pooling layers, and the fully-connected layer is the final layer. Figure 1 shows the architecture of Convolutional Neural Network.

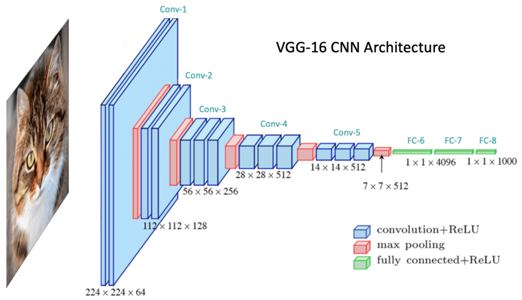

Figure 1. Architecture of Convolutional Neural Network

The VGG-16 network is specifically used to accept color images with an input shape of 224x224x3 where the 3 represents the RGB color channels from the example above. As the input data passes through the network, the shape of the data is transformed. The spatial dimensions are (intentionally reduced) while the depth of the data is increased. The depth of data is referred to as the number of channels. Figure 2. Shown the equation to find out the dimension of output when the shape of an output for more complex inputs or filter dimensions.

```
Dimension of image = (n, n)
Dimension of filter = (f,f)

Dimension of output will be ((n-f+1) , (n-f+1))
```

Figure 2. Equation to find dimensions of output for CNN

$$Z = W^T . X + b$$

Figure 3. Equation for linear transformation in fully-connected layer of CNN

Let X is the input, W as weight, and b which named bias is constant, note that the W in this case will be a matrix of (randomly initialized) numbers.

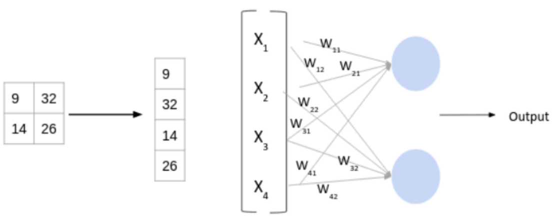

Figure 4. Example to explain equation above

Considering the size of the matrix is (m, n) – m will be equal to the number of features or inputs for this layer. Since we have 4 features from the convolution layer, m here would be 4. The value of n will depend on the number of neurons in the layer. For instance, if we have two neurons, then the shape of weight matrix will be (4, 2)

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \qquad W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \\ W_{41} & W_{42} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Input Data | Randomly Initialized Weight Matrix | Randomly Initialized bias Matrix

Figure 5. Explanation of figure above when weight and bias defined

$$Z = W^T . X + b$$

$$Z = \begin{bmatrix} W_{11} & W_{21} & W_{31} & W_{41} \\ W_{12} & W_{22} & W_{32} & W_{42} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$Z_{2x2} = \begin{bmatrix} W_{11}X_1 + W_{21}X_2 + W_{31}X_3 + W_{41}X_4 \\ W_{12}X_1 + W_{22}X_2 + W_{32}X_3 + W_{42}X_4 \end{bmatrix}$$

Figure 6. Forward propagation process

So far, linear transformation is one of the operation for fully-connected layer on incoming data. Convolutional layer responsible for extracting some valuable features from the data, These features are converted into 1D format and sent to the fully connected layer that generates the final results as fully connected layer only work with 1D data. All of these individual values are treated as separate features that represent the image as well as linear transformation enable CNN to identify and extract important features at different spatial locations.

B. *Method 2: Logistic Regression*

Logistic regression is a statistical model used to predict the probability of a binary outcome based on one or more input variables. It is a type of regression analysis where the dependent variable is categorical or binary. Logistic regression model estimates the parameters that define the logistic function by maximizing the likelihood of the observed data. This is typically done using optimization algorithms such as maximum likelihood estimation or gradient descent.

C. Method 3:SVM

Support Vector Machines (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It is based on the concept of finding an optimal hyperplane that separates different classes in the data. SVM aims to maximise the margin between classes, making it effective for handling both linearly separable and non-linearly separable datasets.

One of the key strengths of SVM is its ability to handle both linear and nonlinear classification problems. By employing different kernel functions, such as linear, polynomial, RBF, or sigmoid, SVM can capture complex decision boundaries. This flexibility makes SVM suitable for a wide range of applications.

SVM also incorporates regularisation through the C parameter, which balances the trade-off between the margin width and misclassifications. A smaller C value emphasises a wider margin, allowing more misclassifications, while a larger C value leads to a narrower margin and fewer misclassifications. Proper tuning of C is essential to prevent overfitting or underfitting.
Overall, SVM is an effective algorithm for classification tasks, providing accurate results and robust performance. Its ability to handle nonlinear problems and find optimal decision boundaries makes it suitable for various domains, including image recognition, text categorization, and bioinformatics.

D. Method 4:KNN

K-Nearest Neighbours (KNN) algorithm is a straightforward and understandable classification approach used for both regression and classification problems. It is a non-parametric approach that depends on how close instances are to one another in the feature space. KNN operates under the presumption that similar cases have related labels.KNN can store the training dataset's feature vectors and combine class labels during the training phase.

KNN has three basic steps.
1. Calculate the distance.
2. Find the k nearest neighbours.
3. Class voting

In other words, KNN calculates the distance between data points, selects the k nearest neighbors, and then assigns a class to the target point based on the majority vote of the neighbors. It is a simple yet effective algorithm for classification tasks, based on the idea that similar data points tend to belong to the same class.

There are 3 calculation formulas that used in KNN, one of which is the Euclidean distance between two points represented by a vector, which can be expressed as follows:

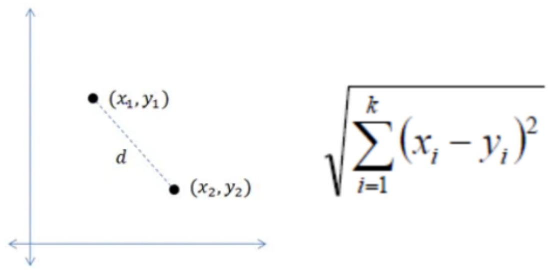

Fig. 1.1.Euclidean distance formula

V. Dataset

We will exclude numbers (1-10) and letters (M, N, S, T) in ASL34 because some of them are dynamic sign language letters and others will mess up our predictions. Therefore, there will be 20 class samples in our collection. Four of us need to perform ASL with the right hand. Then four of us collected 30 samples for each category, so a total of 2400 images were collected.The sample image after preprocessing is shown in Fig. 1.2.

Fig. 2.1.sample image after preprocessing

VI. Experiment

Our experiments are mainly based on three stages. These stages are preprocessing, splitting the dataset and classification using four methods. In this experiment, we capture images of the dataset through a mobile phone or laptop camera. Afterwards, we resized them to reduce computation. In our proposed framework, first we extract some important information from gestures, and then split the images in the database and make the entire dataset be split into 60% training set,20% validation set, and 20% testing set. Classification is done with four algorithms that we mention in methodology.A Proposed Method for One-Handed Sign Language Recognition System.The workflow proposed in the experiment is shown in Fig. 2.1.

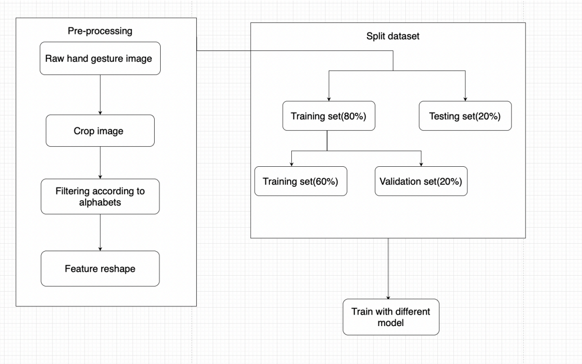

Fig. 2.1 experiment workflow

TABLE I. THE PERFORMANCE OF THE METHODS IN COMPARISON.

Method	**F1-score**	**Precision**	**Recall**	**Accuracy**
CNN	0.84	0.85	0.84	0.85
LR	0.8	0.82	0.79	0.8
SVM	0.87	0.87	0.87	0.87
KNN	0.87	0.87	0.86	0.8667

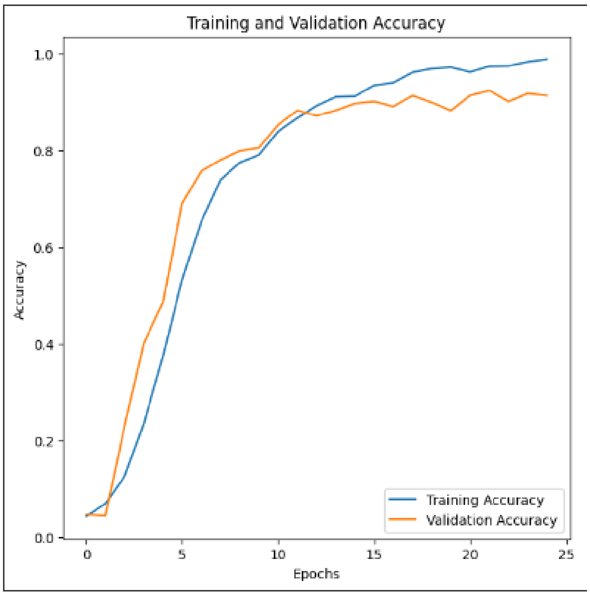

Figure 2.3 Training and validation accuracy for CNN

There are several condition and ways to interpret the training and validation accuracy graph. First, when the training accuracy increases while validation increases in a stable way, it means that the model is performing well with the unseen data and the model is learning well. However, if the training accuracy increase steadily, but the validation accuracy starting to decrease or become plateau like Figure 2.3 above, it means that the model might be overfitting. It can be said that the model is memorizing training data too well but not learning.

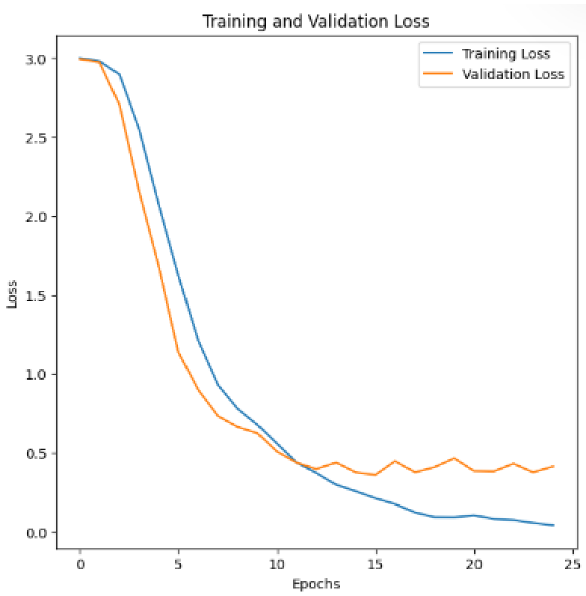

Figure 2.4 Training and validation loss for CNN

Figure 2.4 shows that training loss decreases steadily, yet validation loss decreases unstable, when the validation loss remains high or starts to increase, while the training loss decreases, it might be overfitting for the model. Meaning that this model is unable to generalize well to new examples.

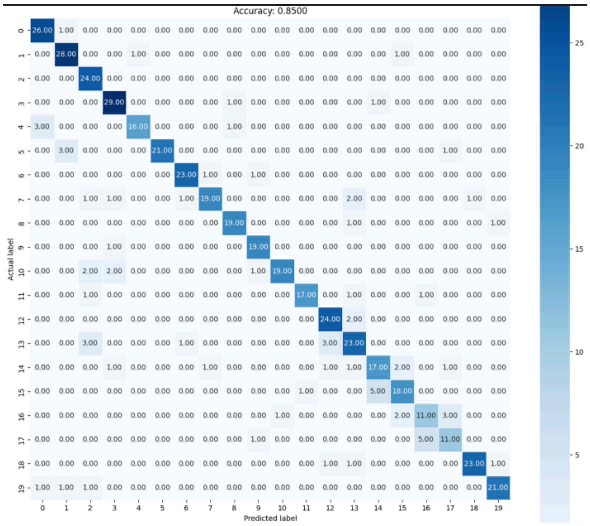

Figure 2.5 Confusion matrix for CNN

Confusion matrix enable us to understand the performance of a model, whether is there any miss classifying for data or which classes have been wrongly classified the most.

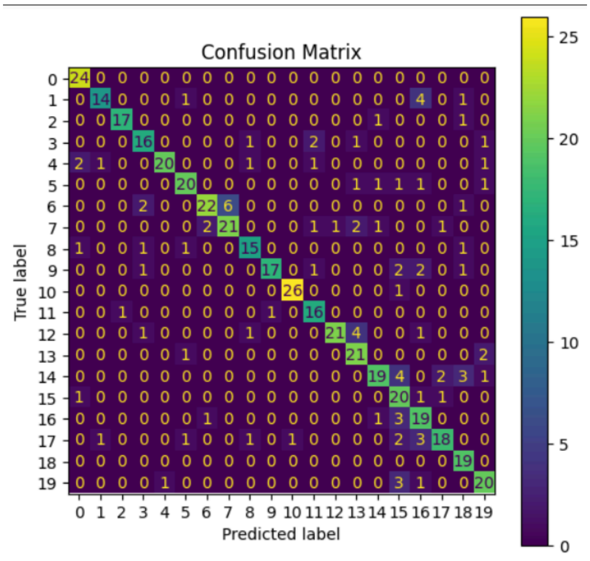

Figure 3.1 Confusion matrix for LR

The confusion matrix is a summary of prediction results on a classification problem. The key of the confusion matrix is used the number of right and wrong predictions which are represented with the count values and broken down by each class. Based on Figure 3.1, the confusion matrix shows the misclassifications made by the model.

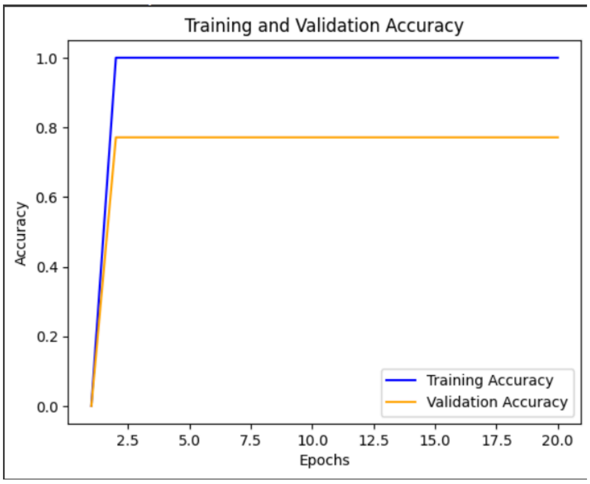

Figure 3.2 Training and validation accuracy for LR

Based on the graph above, we can see that the training accuracy is achieving 1.0 while the validation accuracy only achieves 0.78. This suggests that the model might be overfitting, meaning that the model has memorised the training samples and its label thus causing this model to perform poorly on unseen data.

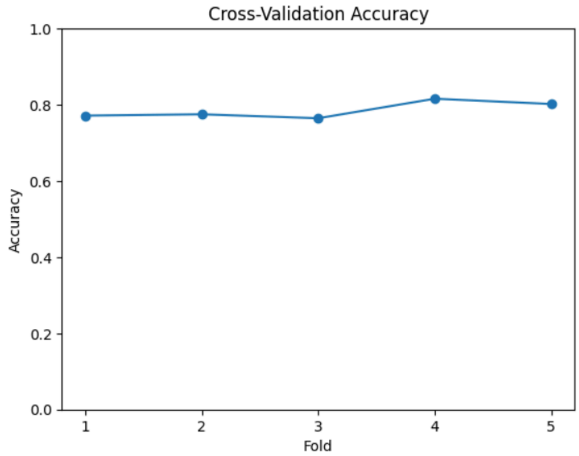

Figure 3.3 Cross-Validation Accuracy for LR

Cross-validation score is the average of the results from each fold. In this case, 5 folds were done and getting an average accuracy of 0.79.

KNN does not have a built-in loss function, the loss is obtained during training of a multi-layer perceptron (MLP) model using the mean squared error (MSE) loss function.So, while KNN does not have an explicit loss function during training, it can still be evaluated using various performance metrics to assess its effectiveness in making predictions.But we still succeed in displaying the loss graph in other specified ways.Below show the KNN training and validation loss graph.

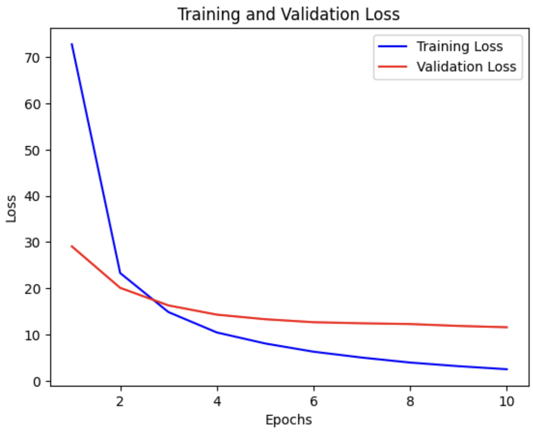

Fig 4.1 KNN Loss graph

KNN has more false positive errors than false negative errors, and we found that KNN tends to misclassify instances as positive (belonging to a class) when in fact they belong to the negative class.Below show the KNN Confusion matrix.

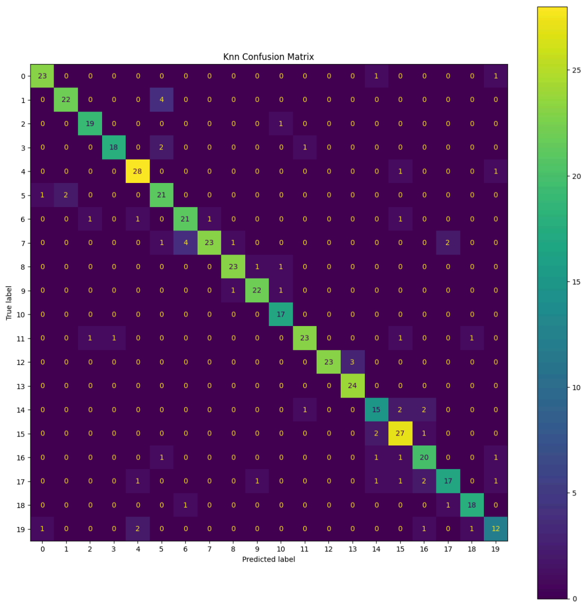

Fig 4.2 KNN Confusion matrix

We use ranges from 1 to 100 for k value to extract and display the mean test scores from the cross-validation results of a grid search . Below shows the table of the mean test score.

TABLE 2. THE MEAN TEST SCORE RESULT

	mean_test_score	std_test_score	params
0	0.823611	0.017208	{'n_neighbors': 1}
1	0.797222	0.018840	{'n_neighbors': 2}
2	0.800000	0.014991	{'n_neighbors': 3}
3	0.792361	0.023178	{'n_neighbors': 4}
4	0.785417	0.018029	{'n_neighbors': 5}
...	...	...	...
94	0.263194	0.025553	{'n_neighbors': 95}
95	0.259722	0.027287	{'n_neighbors': 96}
96	0.255556	0.028683	{'n_neighbors': 97}
97	0.252778	0.024395	{'n_neighbors': 98}
98	0.252083	0.022026	{'n_neighbors': 99}

We use k-Fold Cross-Validation and fold it to 5 to make the dataset will be divided into training and testing 5 times so that each group has a chance to be the test set.Below show the table of the cross validation accuracy.

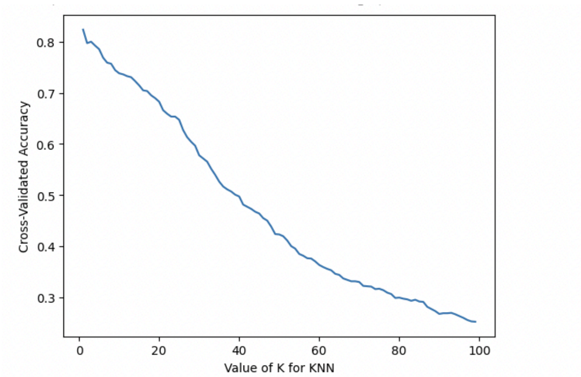

Fig 4.4 KNN Cross validation accuracy

We compare knn by indicating the x-train and the points which matching the k-value as plots to show the different matches of the k-value.

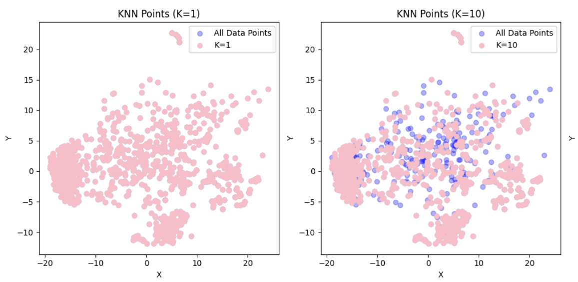

Fig 4.5 KNN points in different k value

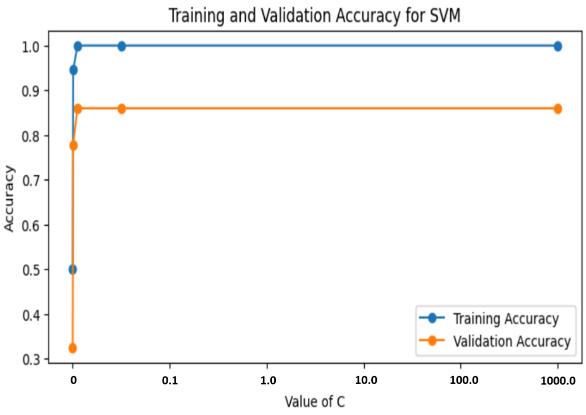

Fig 5.1 Training and validation accuracy for SVM

As the value of C increases, the training accuracy improves significantly, reaching a perfect score of 1.0 for the remaining values of C. The validation accuracy also improves but reaches a plateau at approximately 0.8597 for the last three values of C. This suggests that the SVM model is overfitting the training data for higher values of C, as evidenced by the perfect training accuracy but relatively lower improvement in the validation accuracy.

Parameter Values	Mean Test Score
{'C': 1, 'kernel': 'rbf'}	0.7784722222222222
{'C': 10, 'kernel': 'rbf'}	0.8597222222222222
{'C': 100, 'kernel': 'rbf'}	0.8597222222222222

Fig 5.2 Mean Test Score for Grid Search(SVM)

In order to find out the best parameter values, we have used grid search to explore different combinations of C and kernel values. ‘C’:10,’kernel’:’rbf’ are the best combinations because the model's performance has reached its optimal level when ‘C’= 10 and doesn't improve further with ‘C’=100.

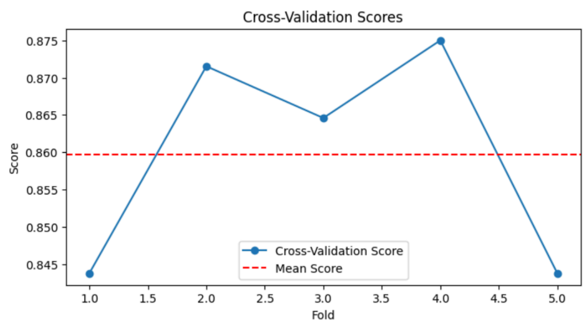

Fig 5.3 SVM Cross-Validation Score

The cross-validation score is obtained by performing cross-validation on the model using the specified number of folds. In this case, cross-validation with 5 folds is performed and the mean cross-validation score of 0.86 suggests that the model has reasonably good performance

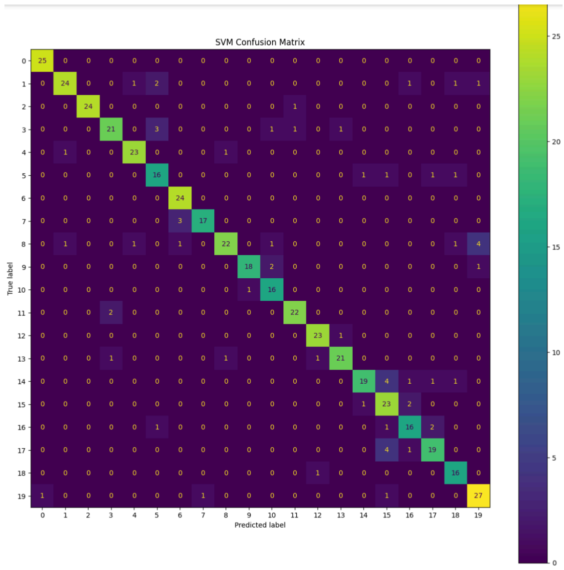

Fig 5.4 SVM Confusion Matrix

Based on the confusion matrix, it appears that SVM has more false positive errors compared to false negative errors. In other words, the model has a tendency to produce "false alarms" predictions.

VII. Conclusion

In this project, we have proposed a machine learning-based American Sign Language (ASL) using a dataset consisting of 2400 images. 4 algorithms have been carried out in this project which are CNN, LR, SVM and KNN for ASL Recognition task.

This project used a dataset consisting of 2400 images excluding numbers (1-10) and letters (M, N, S, T) due to their dynamic nature. All images have been converted to RGB format and resized to (50,50) in the processing step to ensure uniformity and compatibility across the dataset. The dataset was divided into a training set (60%), validation set (20%), and testing set (20%) for evaluation.

Among the methods, SVM and KNN achieved similar performance with an accuracy of 87% and F1-score of 0.87. CNN exhibited slightly lower performance with an accuracy of 85% and F1-score of 0.84. LR showed lower performance compared to other algorithms with 80% of accuracy and F1-score.

Overall, our ASL recognition machine demonstrated promising results,with SVM and KNN showing the highest accuracy in this project underscores the potential of SVM and KNN algorithms in ASL recognition tasks.

VIII. Appendices

Appendices A- Literature review for 12 article paper

1. K. S, M. S, K. R, K. DS and K. M, "Sign Language Recognition using Python and OpenCV," 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 2023, pp. 88-92, doi: 10.1109/ICSMDI57622.2023.00023.
2. G. Singh, A. L. Yadav and S. S Sehgal, "Sign language recognition Using Python," 2022 International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates, 2022, pp. 1-5, doi: 10.1109/ICCR56254.2022.9996001.
3. B. Ben Atitallah et al., "Hand Sign Recognition System Based on EIT Imaging and Robust CNN Classification," in IEEE Sensors Journal, vol. 22, no. 2, pp. 1729-1737, 15 Jan.15, 2022, doi: 10.1109/JSEN.2021.3130982.
4. S.Katoch, V.Singh and U.S.Tiwary, “Indian Sign Language recognition system using SURF with SVM and CNN”, Array14(2022)100141, Computer Science, National Institute of Technology, Hamirpur, India
5. J.Z.Wang, X.Q.Xu and M.Li,“The Study of Gesture Recognition Based on SVM with LBP and PCA”, Journal of Image and Graphics, Vol. 3, No. 1, June 2015, doi: 10.18178/joig.3.1.16-19
6. V.Kumar, R.Narula, A.Ticku, “Hand Sign(ISL) Recognition Model For Alphabets Using CNN Model”,Bharati Vidyapeeth’s College of EngineeringGGSIPU, New Delhi, India,2022 IJRTI | Volume 7, Issue 12 | ISSN: 2456-3315
7. The Study of Gesture Recognition Based on SVM with LBP and PCA - Volume 3, No. 1, June 2015 - Journal of Image and Graphics. (n.d.). http://www.joig.net/index.php?m=content&c=index&a=show&catid=38&id=92
8. IJRTI editor cheif. (n.d.). Welcome to IJRTI. https://www.ijrti.org/viewpaperforall?paper=IJRTI2212041
9. Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14, 100141. https://doi.org/10.1016/j.array.2022.100141

IX. Contribution

MQ carry out the KNN algorithms training in the experiment and participate in the report for introduction,objective,related work,method,dataset experiment and appendices declaration.

Team member Kong Zi Yin explored and applied the Convolutional Neural Network (CNN) algorithm in the experiment. She also participated in the report for introduction, objective, related work, method, dataset experiment and appendices declaration.

Victor was assigned to explore the SVM algorithm in the experiment. He also participated in the report for related work, method, dataset experiment, appendices declaration and conclusion.

Wei Lun is responsible for the Logistic Regression algorithm in the experiment. He participated in the report for related work, method, and dataset experiment.

X. Acknowledgement

The authors would like to thank our senior lecturer Ts. Dr Lim Kian Ming, Multimedia University Melaka provided guidance and support for this work.

XI. Reference

1. P. Haque, B. Das and N. N. Kaspy, "Two-Handed Bangla Sign Language Recognition Using Principal Component Analysis (PCA) And KNN Algorithm," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-4, doi: 10.1109/ECACE.2019.8679185.

2.N. E. A. Rashid, Y. A. I. M. Nor, K. K. M. Sharif, Z. I. Khan and N. A. Zakaria, "Hand Gesture Recognition using Continuous Wave (CW) Radar based on Hybrid PCA-KNN," *2021 IEEE Symposium on Wireless Technology & Applications (ISWTA)*, Shah Alam, Malaysia, 2021, pp. 88-92, doi: 10.1109/ISWTA52208.2021.9587404.

3.M. Fora, B. Ben Atitallah, K. Lweesy and O. Kanoun, "Hand Gesture Recognition Based on Force Myography Measurements using KNN Classifier," *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*, Monastir, Tunisia, 2021, pp. 960-964, doi: 10.1109/SSD52085.2021.9429514.

7. Deep Learning for Image Classification in python with CNN. ProjectPro. (n.d.). https://www.projectpro.io/article/deep-learning-for-image-classification-in-python-with-cnn/418#toc-7

8. Convolutional Neural Network (CNN) : Tensorflow Core. TensorFlow. (n.d.). https://www.tensorflow.org/tutorials/images/cnn

9. Convolutional Neural Networks (cnns) and layer types. PyImageSearch.https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

10. Tolentino, L. K. S., Juan, R. S., Thio-ac, A. C., Pamahoy, M. A. B., Forteza, J. R. R., & Garcia, X. J. O. (2019). Static sign language recognition using deep learning. *International Journal of Machine Learning and Computing*, *9*(6), 821-827.

11. More, V., Sangamnerkar, S., Thakare, V., Mane, D., & Dolas, R. (2021). Sign language recognition using image processing. JournalNX, 85-87.

12. Kaur, H., Bedi, S. K., & Lekhana, M. A. (2023). Image-based Indian Sign Language Recognition: A Practical Review using Deep Neural Networks. *arXiv preprint arXiv:2304.14710*.

13. Gibiansky, A. (n.d.). Convolutional Neural Networks - Andrew Gibiansky. https://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/#:~:text=Convolutional%20Layers,-Suppose%20that%20we&text=In%20order%20to%20compute%20the,)(j%2Bb).

14. Singh, A. (2020). Demystifying the Mathematics Behind Convolutional Neural Networks (CNNs). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/#:~:text=There%20is%20a%20simple%20formula,part%20of%20the%20CNN%20architecture.