International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Modified 3D-Convolutional Neural Network Applied in Dynamic Filipino Sign Language Recognition

Erika Bisoy[1], Chriszel Abby Oduca[2]

[1,2] Student, College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila

ABSTRACT

Advancing communication for the deaf and mute community requires effective sign language recognition systems. This study improves dynamic gesture recognition for Filipino Sign Language (FSL) using a 3D Convolutional Neural Network (3D CNN). Key challenges such as vanishing and exploding gradients, which hinder the model's learning capabilities, were addressed through batch normalization and gradient clipping. Batch normalization stabilized training by reducing gradient variance from 366.8836 to 4.9723. Gradient clipping minimized instability, leading to increased model accuracy. These techniques significantly enhanced model robustness and generalization. This research highlights their importance in developing accessible FSL recognition systems that promote inclusivity and preserve language.

Keywords: Filipino Sign Language, 3D Convolutional Neural Network, Machine Learning

INTRODUCTION

Dynamic gesture recognition is essential for understanding and classifying hand movements in video frames, enabling communication for the deaf community. Filipino Sign Language (FSL), recognized by Republic Act No. 11106, embodies Filipino culture and syntax. However, challenges like inefficient learning materials and misconceptions hinder accessibility. This study focuses on improving dynamic gesture recognition for FSL, addressing the social isolation of 1.78 million Filipinos with hearing difficulties. 3D CNNs, particularly suited for video analysis, integrate spatiotemporal data for effective gesture recognition. These networks utilize convolutional layers to extract spatial and temporal features, pooling layers to reduce computational complexity, and activation functions like ReLU to capture non-linear patterns. Fully connected layers finalize the classification process. By leveraging these capabilities, the study aims to enhance gesture recognition performance. Sign language, a visual medium combining gestures and expressions, plays a critical role in communication. Despite the implementation of the FSL Act, inadequate resources limit deaf learners' access to education and opportunities. Addressing these gaps, this study explores 3D CNN enhancements to foster inclusivity and societal integration.

RELATED LITERATURE

In **"A Summary of Literature Review: Convolutional Neural Networks"** (Celine, et. al, December

2021), CNN is the subset of neural networks that are used to handle organized grid-like data including a time series and even photographs. These models are especially well-suited for carrying out tasks such as picture classification, object identification, and audio recognition, because of the effectiveness with which their design captures hierarchical patterns and characteristics within the datasets. CNN has been responsible for the development of innovation that transcends several fields: computer vision and image processing. The challenge of improving CNN for complex visualization models highlights the need for optimizing algorithms that are capable of producing reliable results over a broad variety of datasets and topologies.

In the article titled **"Sign Language Recognition Using Convolutional Neural Network,"** the author focuses on developing a computer vision-based application that translates sign language into text using a Convolutional Neural Network (CNN). This application aims to bridge the communication gap between signers and non-signers. Nandhini et al (2021) suggested the use of filters in the sign language translation algorithm because the existing system has low accuracy as it faced issues with skin tone identification. Sign language conversion can reach a maximum of 96% accuracy but achieving that can be slow. The current system failed to obtain this accuracy as it lagged in identifying the skin tone under the low-light areas. The key goal of this study is to recognize the sign with maximum accuracy apart from different light, dark conditions must be developed. The proposed approach uses Convolutional Neural Networks (CNNs), a type of deep learning model known for its efficiency in image and video processing tasks. By training the CNN on images of various sign language gestures, the system can accurately translate these gestures into corresponding text. Overall, the proposed system successfully predicts the signs of signs and some common words under different lighting conditions and different speeds. Accurate masking of the images is being done by giving a range of values that could detect human hands dynamically. The proposed system uses CNN for the training and classification of images. The system gets better at recognizing signs by analyzing many images (1750 per sign) and extracts key details. It can then show the recognized word as text and even speak it out loud. With support for 125 words including alphabets, this user-friendly system is accessible for deaf and hearing people alike.

The **"A Review on Sign Language Recognition using CNN"** by Ugale et. al (2023) indicates that the problem with manual sign language intrudes on the individual right to privacy. This can be resolved by using an automated sign language translator that can translate sign language into spoken or written language. CNN is a suitable tool for simulating sign language recognition. A CNN is a type of network design for deep learning algorithms utilized for tasks, image recognition and pixel data processing. This design employs Convolutional layers with various pooling sizes, an activation function, and a rectified linear unit handling non-linearities. With supervised learning, the network is trained to learn the characteristics of each symbol. The researchers suggest a virtually identical architecture, including image Augmentation, feature extraction, and fully connected layers.

From the article of Cisco (n.d) at Linkin titled, **"What are the advantages and disadvantages of using convolutional neural networks for image recognition"**, CNN can learn from raw pixel data, this means that they can automatically discover and adapt to the most salient characteristic of the images; edges, shapes, colors, textures, and objects. The inherent bias data can be learned with fewer examples compared to some vision transformer architectures. CNN requires a large amount of labeled data to train effectively which is causing costly and time-consuming to obtain and annotate. They are prone to overfitting so that in the process they can memorize the noise and details training data and fail to generalize to new and different data. In image recognition CNN is not only an option, but these also

include Recurrent Neural Networks (RNN) for neural network architecture, Capsule Networks (CapsNets) that model the semantic relationships between two objects, and Generative Adversarial Networks (GANs) that is an artificial neural network that can generate realistic and diverse images from random noise. Each of these offers unique advantages in application. With the CNN structure, several layers, activation functions, and normalization make the hyperparameter space exceedingly large. In the CNN layer, convolutional is very helpful in enabling compression in neural network weights, this helps to save memory and computational resources, and this enables the network to generalize better to new data, improving its overall efficiency and performance.

From the article **"Exploring Convolutional Neural Network: Architecture, Steps, Use Cases, and Pros and Cons"** by Tamanna 2023, published on medium website, CNN is inspired by the biological visual cortex that is responsible for processing visual information in animals. The main core of CNN is its ability to automatically learn relevant features from raw input data that is effective in image classification, object detection, and image segmentation. CNN uses a series of convolutional Layers to extract features from input images. There are multiple layers that are involved in building CNN, input layer, convolutional layer, activation function, pooling layer, and fully connected layer. CNN has a wide range of use cases; image classification for detecting faces, objects, and handwritten, object detection, image segmentation for separating foreground and background, and medical image analysis for analysis tasks such as tumor detection, and image registration. The advantages of CNN to other traditional machine learning algorithms are it can automatically learn relevant features from raw data, has high accuracy, and is robust to noise, and is highly effective in real-world applications. CNN also has a disadvantage such as it requires significant high-end specs to train and deploy, it requires large amounts of training data to have high accuracy, and prone to overfitting when the training Data is limited and noisy, which can lead to poor generalization performance on new data. With the ability of CNN to automatically learn relevant features from raw input data this makes them highly effective in a wide range of applications.

METHODOLOGY

The methodology for this study focuses on developing and evaluating a modified 3D Convolutional Neural Network (3D CNN) for dynamic Filipino Sign Language (FSL) recognition. The research begins with the design and enhancement of a 3D CNN framework tailored to capture spatiotemporal patterns in dynamic gestures. The model integrates batch normalization to stabilize gradient magnitudes and gradient clipping to prevent exploding gradients, addressing common challenges in deep learning. The conceptual design is informed by existing algorithms, such as the Naik and Soni framework, which serve as a foundation for feature extraction and classification. The training phase incorporates key enhancements, including batch normalization to mitigate gradient instability, gradient clipping to prevent large weight updates, and dropout regularization to address overfitting. Features are extracted using 3D convolutional layers, followed by classification through dense layers with dropout applied. Model training is conducted using the model.fit() method, optimizing stability and convergence with these enhancements. Comparisons between the baseline and enhanced 3D CNN models highlight the effectiveness of the proposed techniques in improving gradient stability and accuracy. Tools like Python, TensorFlow, OpenCV, and MediaPipe are used for implementation, supported by a high-performance hardware setup and software like Anaconda and PyCharm. This methodology establishes a robust framework for enhancing the 3D CNN's performance in dynamic FSL recognition, addressing key

challenges like vanishing and exploding gradients while ensuring accuracy and stability.

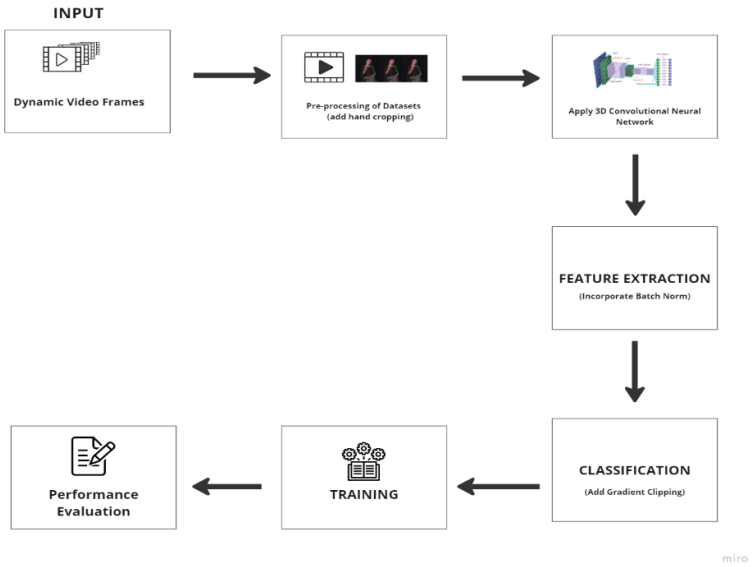


Figure 1 shows the stage process for dynamic video frame classification.

PROPOSED ENHANCED ALGORITHM

Enhanced Algorithm 2. Applying 3D CNN Architecture

Input: Array of frames from videos

Output: Accuracy

1. Feature Extraction:

Decide the following parameters -

i. The kernel/Filter - usually 3*3 matrix

ii. BatchNormalization()

iii. The filter count - how many filters to use

iv. Stride - steps of the filter

v. Padding - the layer of 0-value pixels

vi. Dropout() - apply dropout after specific layers (e.g., 30% for convolutional layers)

2. Classification

Extracted features are converted to class

i. Dropout() - apply dropout to dense layers (e.g., 50%)

3. Training using model.fit() method

i. Define the optimizer with gradient clipping:

- use ‘clipnorm’ or clipvalue parameter in the optimizer

4. Try model 5 and 50 epochs

Performance Evaluation

Enhanced Algorithm 1. Data Pre-Processing

Input: Videos in AVI file format, FRAME_COUNT, FRAME_HEIGHT, FRAME_WIDTH.

Output: Array of cleaned and processed frames.

1.Read the directory:

listing = os.listdir(CLIPS_DIR)

2. For each video file i in listing:

i. Open the video using OpenCV:

cap = cv2.VideoCapture(i)

Determine the total number of frames:

total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

ii. Uniform Sampling:

Select evenly spaced frame indices:

frame_indices = np.linspace(0, total_frames - 1, FRAME_COUNT, dtype=int)

iii. Resize Frames:

Resize each frame to a fixed size:

frame = cv2.resize(frame, (FRAME_WIDTH, FRAME_HEIGHT))

iv. Normalize Pixel Values:

Scale pixel values to the range [0, 1]:

frame = frame / 255.0

v. Append Processed Frames:

Append the processed frame to the list of frames.

vi. Handle Videos with Fewer Frames:

If the total number of frames is less than FRAME_COUNT, pad the frames with zero-filled frames:

pad_frame = np.zeros((FRAME_HEIGHT, FRAME_WIDTH, 3), dtype=np.float32)

3. Convert the List of Frames:

Convert the list of frames into an array and append it to the dataset.

4. Repeat Steps 1–3 for all six action classes.

5. Call Algorithm 2 for further processing.

The proposed enhanced algorithms focus on optimizing the performance of a 3D Convolutional Neural Network (CNN) for dynamic gesture recognition, particularly tailored for Filipino Sign Language. This enhancement comprises two main parts: data pre-processing and the application of the 3D CNN architecture. **Data Pre-Processing:** This phase involves preparing video data for subsequent processing. Videos stored in AVI format are first accessed from a specified directory. For each video, the algorithm captures frames, determining the total number of frames to process. A uniform sampling technique is applied to select evenly spaced frames, which are then resized to standardized dimensions and normalized to scale pixel values between 0 and 1. This standardization is crucial for maintaining consistency across all inputs. For videos with fewer frames than needed, padding is applied using zero-filled frames to ensure uniformity in input size and shape. **Applying 3D CNN Architecture:** Once the frames are pre-processed, they are fed into the enhanced 3D CNN. This part of the algorithm begins with feature extraction, where several parameters are set, including the size of the kernels (filters), the number of filters, stride, and padding. Batch normalization is employed to stabilize the network during training by normalizing the inputs of each layer. Dropout is strategically applied after specific layers to prevent overfitting, ensuring that the network generalizes well to new, unseen data. The classification step then processes these features to produce output classes. Training is conducted using the model.fit() method, where the optimizer is carefully configured with gradient clipping to manage the magnitude of updates

and prevent training instabilities. The training process tests the model across various epochs to optimize performance and assess generalization capabilities. **Performance Evaluation:** After training, the model's performance is evaluated to determine its accuracy in recognizing and classifying dynamic gestures accurately. This evaluation helps in fine-tuning the model to achieve the best possible performance in real-world applications.This approach ensures a robust training regime that enhances the model's ability to interpret Filipino Sign Language accurately, aiming to improve assistive communication technologies.

RESULTS AND DISCUSSION

The proposed enhancements to the 3D CNN algorithm focus on addressing key challenges such as vanishing and exploding gradients. These issues were mitigated through the integration of batch normalization and gradient clipping. Batch normalization stabilized the training process by normalizing inputs across each layer, ensuring consistent gradient magnitudes, while gradient clipping limited excessively large gradients, preventing disruptive updates during optimization. These techniques collectively improved training stability, convergence speed, and overall model performance.

A. Gradient Stability

The stability of gradient norms was a critical focus of the study. Figures 2 and 3 illustrate the gradient norms over epochs for the baseline and enhanced models. The baseline (non-batch normalized) model exhibited significant fluctuations and instability, with a variance of 366.8836. In contrast, the enhanced model with batch normalization showed consistent and stable gradient behavior, reducing the variance to just 4.9723 (Figure 3). This reduction highlights the effectiveness of batch normalization in ensuring smooth optimization and mitigating exploding gradients.

Figure 2: Gradient Norms over Epochs - Non-Batch Normalized (5 epochs)

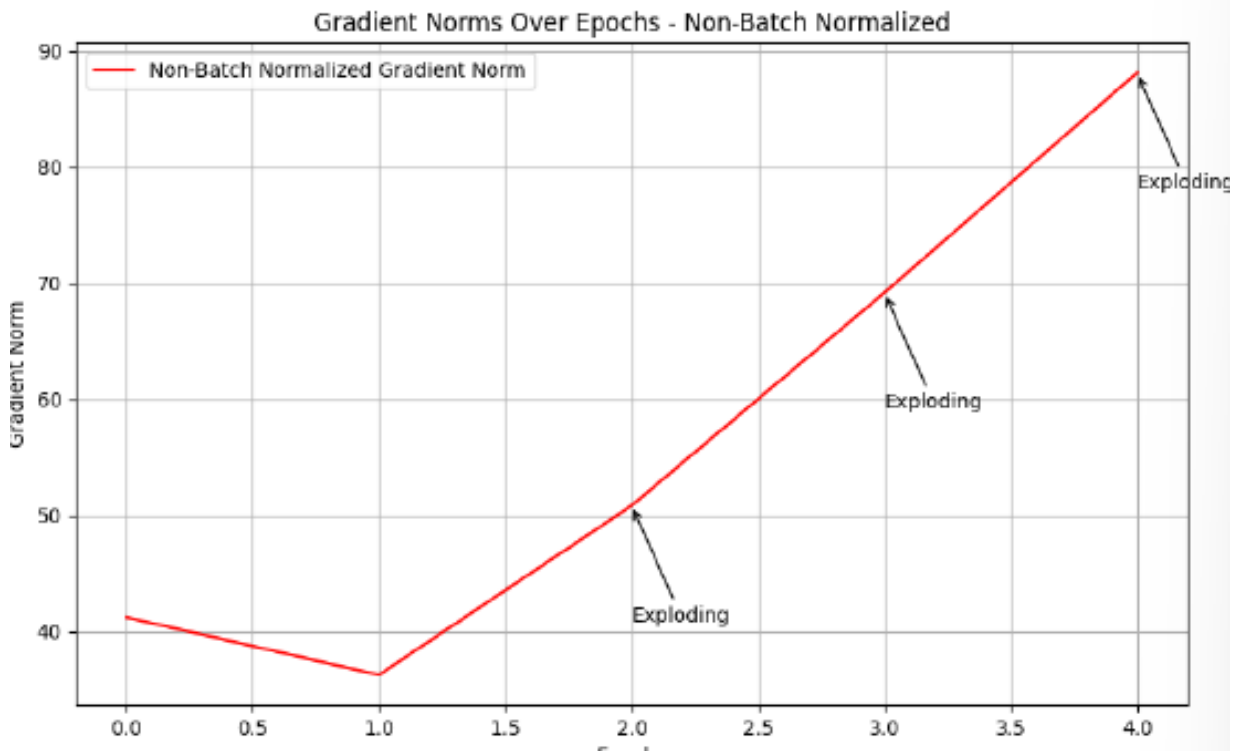


Figure 3: Gradient Norms over Epochs - Batch Normalized (5 epochs)

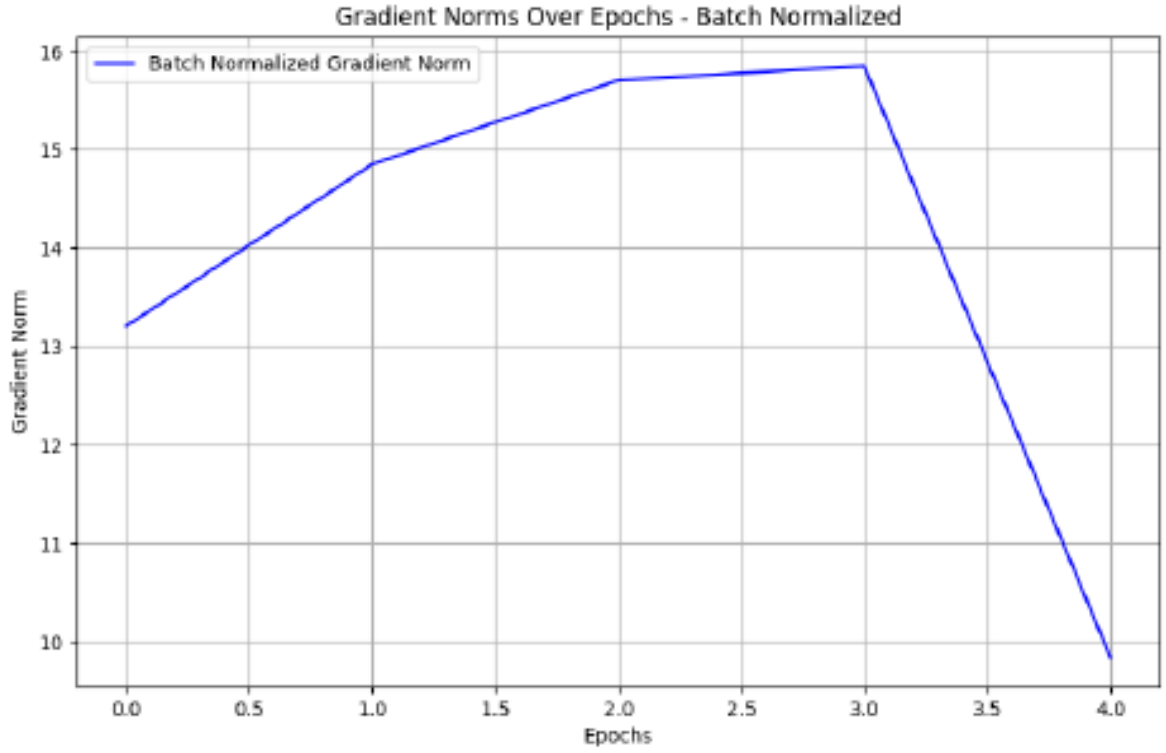


Table 1: Key Metrics and Observations (5 Epochs)

Metric	Non- Batch Normalized	Batch Normalized
Variance of Gradient	366.8836	4.9723
Stability of Training	Unstable (Exploding Gradients)	Stable
Optimization Efficiency	Low	High

The variance of gradient norms in the non-batch normalized model was notably high at 366.8836, indicating instability in the gradient flow, which likely contributed to issues such as exploding gradients. In contrast, the batch-normalized model showed a much lower variance of 4.9723, demonstrating that batch normalization effectively stabilized the gradient magnitudes. This stabilization directly improved the stability of training, with the non-batch normalized model being unstable due to the presence of exploding gradients. In contrast, the batch-normalized model exhibited stable training conditions. Moreover, optimization efficiency was considerably lower in the non-batch normalized model, as gradient instability hindered effective learning and slowed convergence.

B. Accuracy and Loss Trends

Figures 4 and 5 compare the training and validation accuracy between the baseline and enhanced models. The baseline model achieved moderate accuracy improvements but suffered from a large gap between training and validation accuracy, suggesting overfitting. On the other hand, the enhanced model demonstrated a steady and significant increase in accuracy for both training and validation sets, with a smaller gap between the two curves, indicating better generalization. Validation loss trends further support this improvement, as seen in Figure 5, where the enhanced model exhibited consistent reductions in loss compared to the baseline model.

Figure 4: Accuracy with Non-Batch Normalization (5 Epochs)

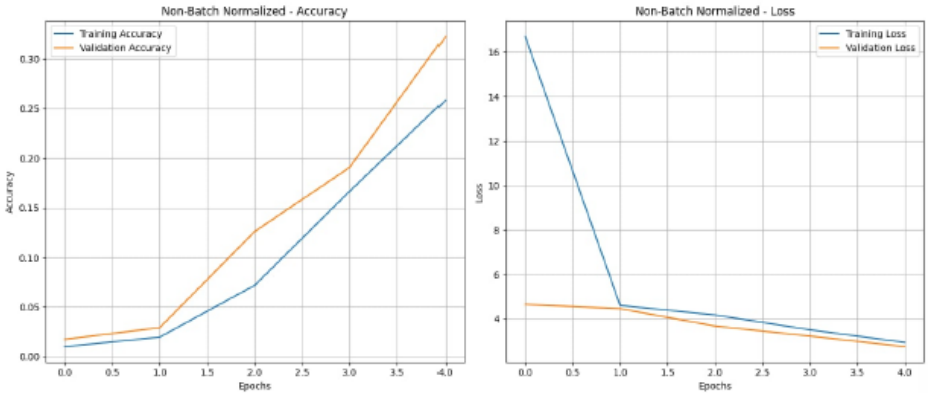
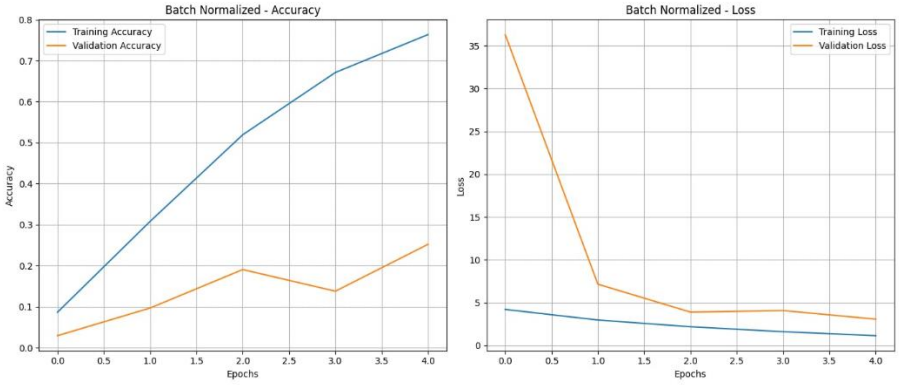


Figure 5: Accuracy with Non-Batch Normalization (5 Epochs)



C. Overall Model Performance

The enhanced 3D CNN model outperformed the baseline in terms of stability, accuracy, and generalization. The incorporation of gradient clipping addressed exploding gradients, while batch normalization improved convergence speed and training efficiency. The combined techniques enabled the model to effectively recognize dynamic Filipino Sign Language gestures, addressing the critical challenges identified in the baseline approach.

CONCLUSION AND RECOMMENDATIONS

The results demonstrated the effectiveness of integrating batch normalization and gradient clipping in enhancing 3D CNN performance. Gradient stability improved significantly, with variance reduced from 366.8836 to 4.9723, and accuracy trends confirmed better generalization. However, to further address overfitting issues, techniques such as dropout regularization, weight decay, and early stopping were explored. These methods helped prevent the model from over-relying on specific neurons, reduced co-adaptation, and ensured training was halted before the model began to overfit the training data. These additional strategies complemented the enhancements, resulting in a more robust and generalized model for dynamic Filipino Sign Language recognition.

REFERENCES

1. Alamuri, M. (2014). A survey of distance/similarity measures for categorical data. *Semantic Scholar*. Retrieved from [https://www.semanticscholar.org/paper/A-survey-of-distance-similarity-

measures-for-data-Alamuri-Bapi/7d2e8177da35ebe768ea86481ab1114b3d6233ee](https://www.semanticscholar.org/paper/A-survey-of-distance-similarity-measures-for-data-Alamuri-Bapi/7d2e8177da35ebe768ea86481ab1114b3d6233ee)

2. Republic Act No. 11106: Filipino Sign Language Act. (2018). *The LawPhil Project*. Retrieved from https://lawphil.net/statutes/repacts/ra2018/ra_11106_2018.html
3. Tech Science. (2023). CNN Injected Transformer for Image Exposure Correction. Retrieved from https://www.techscience.com/cmc/v70n3/44942/html
4. Springer. (2023). Human cognition dynamics through neural networks. Retrieved from https://link.springer.com/article/10.1007/s12559-023-10177-w
5. Garcia, B., & Viesca, S. A. (2016). Real-time American Sign Language recognition with convolutional neural networks. *Stanford University*.
6. Tamanna. (2023, April 24). Exploring convolutional neural networks: Architecture, steps, use cases, and pros and cons. *Medium*. Retrieved from https://medium.com/@tam.tamanna18/exploring-convolutional-neural-networks-architecture-steps-use-cases-and-pros-and-cons-b0d3b7d46c71
7. Yan, J. (2023). Application of CNN in computer vision. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/377826786_Application_of_CNN_in_computer_vision
8. Sheth, P., Dedhia, R., Chheda, A., & Sawant, V. (2023). American sign language recognition and generation: A CNN-based approach. Retrieved from https://www.academia.edu/105216952/American_Sign_Language_Recognition_and_Generation_A_CNN_based_Approach
9. Borad, A. (2024, April 30). Regularization: Make your machine learning algorithms "learn," not "memorize." *eInfochips*. Retrieved from https://www.einfochips.com/blog/regularization-make-your-machine-learning-algorithms-learn-not-memorize/
10. OneNews. (2023). Why you should start learning sign language, an important life skill. Retrieved from https://www.onenews.ph/articles/why-you-should-start-learning-sign-language-an-important-life-skill-1
11. Department of Science and Technology - Science and Technology Information Institute (DOST-STII). (2023). DOST helps promote inclusivity through the use of Filipino sign language. Retrieved from https://www.stii.dost.gov.ph/1762-science-helps-promote-inclusivity-through-use-of-filipino-sign-language

12. Naik, K. J., & Soni, A. (2021). Video classification using 3D convolutional neural network. *ResearchGate*. Retrieved from https://www.researchgate.net/publication/348119792_Video_Classification_Using_3D_Convolutional_Neural_Network
13. Sartori, P. (2021). Comparison of 2D and 3D convolutional neural networks for medical imaging semantic segmentation. *M.S. thesis, Department of Mathematics, Politecnico di Milano*.
14. Wang, H., Pu, G., & Chen, T. (2022). A lip reading method based on 3D convolutional vision transformer. *IEEE Access, 10*, 84367–84378. https://doi.org/10.1109/ACCESS.2022.3193231
15. Linguistics UPD. (2023). Liberty Notarte Balanquit presents Filipino Sign Language research. *University of the Philippines Linguistics Department*. Retrieved from https://linguistics.upd.edu.ph/news/liberty-notarte-balanquit-presents-filipino-sign-language-research/
16. Hayou, S., Doucet, A., & Rousseau, F. (2021). On the impact of the activation function on deep neural networks training. *Proceedings of the 38th International Conference on Machine Learning, 119*, 5381–5391. Retrieved from https://arxiv.org/abs/1902.06853
17. Noroozi, M., Pirsiavash, H., & Favaro, P. (2017). Motion deblurring using CNNs. *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 57–65. Retrieved from https://arxiv.org/abs/1701.01486
18. Toledo, R., Garcia-Garcia, A., Orts-Escolano, S., & Cazorla, M. (2019). Overfitting and its impact on convolutional neural networks: A review. *Pattern Recognition Letters, 128*, 173–182. https://doi.org/10.1016/j.patrec.2019.08.005