Convex Optimization for Feature Refinement in CNN-Based Hand Gesture Classification

Huairu Chen
Department of ECE
University of Toronto
Toronto, Canada
huairu.chen@mail.utoronto.ca

Shiming Zhang
Department of ECE
University of Toronto
Toronto, Canada
shim.zhang@mail.utoronto.ca

***Abstract*—Deep convolutional neural networks (CNNs) have become the standard for image-based classification tasks, including hand gesture recognition. However, standard CNNs often struggle with noisy, redundant, or unstructured intermediate features that may reduce classification performance, especially under challenging conditions. Traditional signal processing methods like sparse coding [1] and total variation (TV) regularization [2] have proven effective in addressing these issues, but integrating them into modern CNNs remains underexplored. In this work, we introduce a hybrid deep learning model that integrates convex optimization techniques—namely sparse coding via FISTA [7] and total variation regularization—into the CNN architecture to refine feature representations. We formulate sparse feature reconstruction using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) and incorporate TV as a spatial smoothness constraint on feature maps. Our methodology enhances internal feature quality by enforcing both sparsity and spatial continuity. The result is a CNN architecture with improved generalization and robustness for hand gesture classification tasks.**

***Index Terms*—CNN, sparse coding, total variation, convex optimization, FISTA, hand gesture recognition**

I. Introduction

Hand gesture classification plays a key role in applications such as sign language interpretation, virtual reality interfaces, and robotics. Recent advances in deep learning, especially CNNs, have made it possible to automatically extract features and classify gestures with high accuracy. However, CNNs still face important challenges. First, their internal feature representations often include spatially inconsistent or noisy activations, especially in real-world conditions with lighting variation, occlusion, and cluttered backgrounds. Second, deep models are often treated as black boxes, lacking interpretability and regularized control over learned representations.

In traditional signal processing and statistical learning, these problems are commonly addressed using sparse coding and regularization techniques. Sparse coding emphasizes the use of a small number of active elements to represent signals, promoting compactness and robustness. Total variation regularization is used to reduce noise while preserving critical spatial structure such as edges.

These ideas can be reintroduced into CNNs through the lens of convex optimization. Instead of relying exclusively on backpropagation, we propose introducing sparse coding and total variation modules as embedded, differentiable components within the network. This allows us to interpret feature refinement as solving a convex subproblem at each layer, combining deep learning's representational power with the robustness and interpretability of convex optimization.

II. Background

The integration of convex optimization techniques such as Sparse Coding and Total Variation (TV) Regularization into CNN architectures proposes a compelling method to address some of the innate challenges faced by standard CNNs in complex visual recognition tasks.

A. Importance of Sparse Coding in CNNs

Sparse Coding aims to reconstruct signals using the fewest possible active elements from a learned dictionary, promoting feature sparsity. In CNNs, introducing sparsity can significantly reduce the redundancy in learned feature representations, which typically leads to more efficient models that are less prone to overfitting and better at generalizing from limited training data. Moreover, sparse representations are often more interpretable, a desirable attribute in critical applications. Papyan *et al.* [1] formalised this view by analysing CNNs through the lens of convolutional sparse coding.

B. Advantages of Total Variation Regularization

Total Variation (TV) Regularization is a technique used extensively in image processing to enhance image quality by reducing noise while preserving edges. When applied to CNNs, TV regularization helps in smoothing the feature activations across spatial dimensions, which not only helps in reducing training noise but also preserves important structural details in the visual input. This regularization is particularly beneficial for tasks involving detailed or fine-grained visual distinctions, such as hand gesture recognition, where edge details are crucial for accurate classification.

C. Literature Review

Recent advancements in integrating convex optimization techniques into deep learning have shown promising results across various domains. Notable studies include the work by Gregor and LeCun [6], who introduced the Learned Iterative

Shrinkage-Thresholding Algorithm (LISTA) that accelerates the sparse coding process within deep networks. Their approach demonstrated that networks embedding LISTA could learn faster and achieve better sparse representations compared to traditional methods.

Another significant contribution is by Beck and Teboulle [7] in the development of FISTA, which improves the efficiency of solving L1-regularized problems commonly found in image processing tasks. Their method has been adapted into deep learning frameworks to enhance feature extraction by promoting sparsity directly within network layers, which has been shown to improve classification accuracy in tasks such as image recognition and signal processing.

Compared to these approaches, our method leverages both Sparse Coding via FISTA [7] and Total Variation (TV) Regularization [2], [9] to not only enhance sparsity but also ensure spatial continuity in feature maps. This dual approach is particularly beneficial for hand gesture recognition where both precision in feature localization and robustness against noise and occlusion are crucial.

III. Methodology

We modify a conventional CNN used for hand gesture classification by embedding two key modules:

- A FISTA-based sparse coding layer inserted after the second convolutional block.
- A total variation regularization loss term added to the training objective.

The sparse coding layer replaces raw convolutional features with sparsely reconstructed alternatives, while the TV term encourages spatial continuity in the refined features.

IV. Mathematical Formulation

A. Sparse Coding via FISTA

To refine noisy or redundant CNN feature representations, we adopt a sparse coding approach that reconstructs each feature vector using a small number of dictionary atoms. Let $\mathbf{f} \in \mathbb{R}^n$ be a feature vector extracted from a CNN layer at a given spatial location. The goal is to find a sparse code $\mathbf{x} \in \mathbb{R}^k$ such that $\mathbf{f} \approx D\mathbf{x}$, where $D \in \mathbb{R}^{n \times k}$ is an overcomplete dictionary with $k > n$.

This leads to the following convex optimization problem:

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{f} - D\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\}, \tag{1}$$

where λ is a regularization parameter that controls the trade-off between reconstruction accuracy and sparsity.

FISTA Optimization: The Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) offers an efficient way to solve the sparse coding problem by incorporating momentum from previous iterations to accelerate convergence.

Gradient Step:

$$\mathbf{z}^{(t)} = \mathbf{y}^{(t-1)} - \alpha D^\top (D\mathbf{y}^{(t-1)} - \mathbf{f}), \tag{2}$$

where α is the step size, typically set to $1/L$, with L being the Lipschitz constant of the gradient of the objective function.

Shrinkage Step (Soft Thresholding):

$$\mathbf{x}^{(t)} = \text{sign}(\mathbf{z}^{(t)}) \odot \max(|\mathbf{z}^{(t)}| - \lambda\alpha, 0), \tag{3}$$

which encourages sparsity in the coefficients by shrinking small values to zero.

Momentum Update:

$$t^{(t)} = \frac{1 + \sqrt{1 + 4(t^{(t-1)})^2}}{2}, \tag{4}$$

$$\mathbf{y}^{(t)} = \mathbf{x}^{(t)} + \left(\frac{t^{(t-1)} - 1}{t^{(t)}} \right) (\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}), \tag{5}$$

where $t^{(t)}$ governs the momentum term to accelerate convergence.

Each spatial location in the feature map is processed independently using this routine. After T iterations, the reconstructed output $\mathbf{f}_{\text{recon}} = D\mathbf{x}^{(T)}$ is used as the refined feature vector passed to subsequent layers.

B. Total Variation Regularization

To encourage spatial smoothness in the learned feature maps while preserving edges, we incorporate Total Variation (TV) regularization. Given a 3D feature tensor $\mathbf{F} \in \mathbb{R}^{C \times H \times W}$ (with C channels, height H, and width W), the TV loss is defined as:

$$\mathcal{L}_{\text{TV}}(\mathbf{F}) = \sum_{c=1}^{C} \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} \left(|\mathbf{F}_{c,i+1,j} - \mathbf{F}_{c,i,j}| + |\mathbf{F}_{c,i,j+1} - \mathbf{F}_{c,i,j}| \right), \tag{6}$$

which penalizes large differences between neighboring spatial locations and promotes piecewise smoothness in the feature maps.

C. Full Loss Function

The final training objective combines standard classification loss with the TV regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(y, \hat{y}) + \alpha \mathcal{L}_{\text{TV}}(\mathbf{F}_{\text{refined}}), \tag{7}$$

where:

- $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss between predicted labels $\hat{y}$ and ground truth labels y,
- $\mathbf{F}_{\text{refined}}$ is the output of the sparse coding layer (used in TV regularization),
- α is a hyperparameter that balances the influence of TV regularization.

This formulation ensures the model not only learns discriminative features for classification but also enforces spatial smoothness and sparsity for improved robustness and generalization.

V. Evaluation of Model Performance

A. Methodology

To evaluate the convolutional neural network's ability to recognize hand gestures, the model was initially trained on a dataset designed for the recognition of twenty-six English alphabets. And thanks to Professor Sinisa Colic, this evaluation

method and model construction is learned from him [11]. Considering the static nature of the intended applications, only nine alphabets (A to I) were selected for training to avoid the complexities associated with motion in other letters. The dataset images featured a hand in front of a clean white background, minimizing environmental noise and ensuring control over experimental variables.

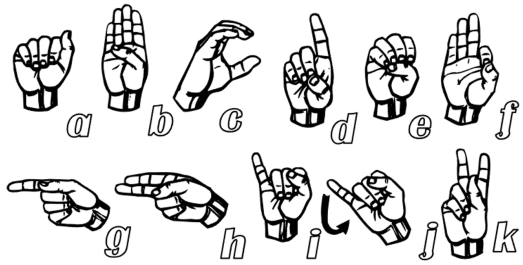


Fig. 1. Example of Hand Gesture Recognition [10]

To test the model under more challenging conditions, additional datasets were sourced from Kaggle [12], featuring hand gestures captured in diverse environmental settings. Which can better capture model's ability to regularize noisy environment.

(a) Clean Data (b) Kaggle Noisy Data

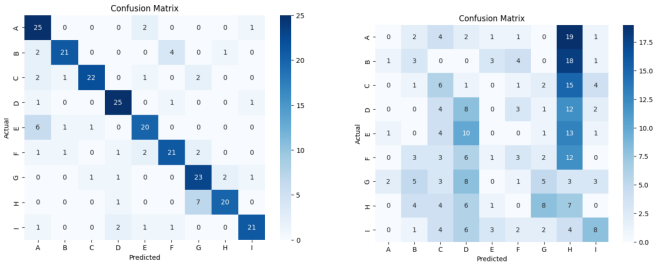
Fig. 2. Data Example

B. Model Architecture and Training

The model was built from scratch using two convolutional layers followed by two fully connected layers, designed to process the static gestures efficiently. After extensive fine-tuning, the model achieved a final validation accuracy of approximately 81.67% and a testing accuracy of 76% on a clean dataset. Kaggle's real-world data set, significantly different from the lab environment, challenged the model's robustness and exposed its limitations, resulting in a drastically lower accuracy of 14.8%.

C. Integration of Sparse Coding and TV Regularization

In response to the initial results, Sparse Coding and Total Variation (TV) regularization were integrated into the model. The Sparse Coding layer, implemented as a custom neural network module, was strategically placed after the second pooling layer to enhance feature extraction and representation. This layer applies a soft thresholding function that introduces



(a) Baseline Test Accuracy 76.0% (b) Kaggle dataset accuracy 14.8%

Fig. 3. Baseline CNN Model

sparsity in the activations, effectively reducing noise and focusing on the most salient features. The soft thresholding is differentiable, which allows the Sparse Coding layer to be seamlessly integrated into the end-to-end training process.

Sparse Coding Implementation: The Sparse Coding layer includes a learnable encoder that projects the input features to a higher-dimensional, sparse space. After encoding, a soft thresholding operation is applied, implemented using a sigmoid function scaled to enforce a sharp transition around a learnable threshold parameter. This process effectively suppresses minor values, enhancing the sparsity of the activations. The sparse representations are then multiplied back with their original encoded projections, ensuring that only significant activations contribute to the layer's output.

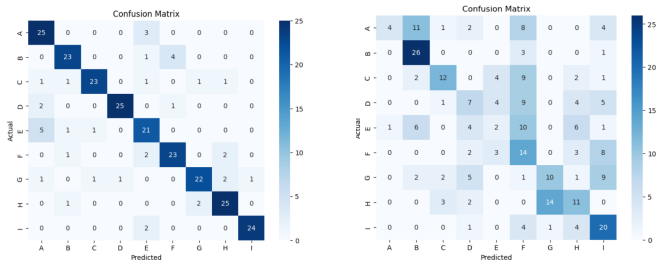
Total Variation (TV) regularization was integrated by augmenting the loss function with a TV-specific loss component. This regularization enforces smoothness in the feature maps, minimizing the total variation across adjacent pixels, which helps in preserving edges while reducing noise. The TV loss is calculated by taking the sum of the absolute differences between adjacent pixels in both horizontal and vertical directions, promoting spatial consistency across the feature maps.

Total Variation Regularization Implementation: The `total_variation_loss` function applies regularization directly to the feature maps produced by convolutional layers. It computes the absolute differences between adjacent pixels along both horizontal and vertical axes, capturing the first-order total variation. The mean of these differences is calculated, providing a scalar that quantifies the amount of variation in the entire feature map. This scalar serves as a regularization term in the overall loss function, guiding the network to produce smoother feature maps while minimizing drastic changes, unless necessary, such as at edges.

By combining these techniques, the model not only minimizes prediction errors but also enhances the quality of the internal representations. The Sparse Coding layer ensures that the network focuses on the most informative parts of the input by enforcing a sparse representation, while the TV regularization maintains the structural integrity of the visual features across different environmental conditions. This dual approach significantly improves the robustness and generalizability of the model, particularly in handling diverse and challenging real-world scenarios.

D. Results and Improvements

The enhancements led to significant improvements in model performance. The test accuracy on the clean dataset increased to 84.4%, while the accuracy on the noisy, real-world data from Kaggle improved to 39.26%. Despite being trained primarily on clean data, the model demonstrated considerable resilience to environmental noise, underscoring the effectiveness of Sparse Coding and TV regularization in managing external variabilities and enhancing generalization.



(a) Model Test Accuracy 84.4% (b) Kaggle dataset accuracy 39.3%

Fig. 4. Updated Model with Sparse Code and TV regularization

E. Conclusion

The integration of advanced regularization techniques Sparse Coding and TV regularization into the CNN architecture markedly improved the model's performance across both controlled and uncontrolled environments. These results highlight the potential of such techniques to significantly enhance the robustness and accuracy of machine learning models in practical applications.

TABLE I
COMPARISON OF MODEL ACCURACIES

Model	Val Acc	Test Acc	Kaggle Acc
Baseline Model	81.67%	76.00%	14.81%
Enhanced Model (Sparse + TV)	83.75%	84.40%	39.26%

VI. DISCUSSION

This study has demonstrated the significant potential of integrating Sparse Coding and Total Variation (TV) Regularization into a convolutional neural network for enhancing hand gesture recognition. The results clearly indicate that these techniques not only improve the model's accuracy on standard test datasets but also markedly enhance its performance in challenging, noisy environments typically encountered in real-world scenarios.

A. Implications of Findings

The enhancements in model robustness and generalizability facilitated by Sparse Coding and TV Regularization are particularly noteworthy. Sparse Coding aids in focusing the model on the most salient features, reducing the influence of noise and irrelevant data. This is critical in applications where precision and reliability are paramount, such as in assistive technology for the visually impaired or in interactive applications like virtual reality. Meanwhile, TV Regularization promotes the smoothness of the feature maps, preserving essential structural details that are crucial for accurately interpreting complex gestures.

B. Limitations

Despite these advancements, the study faces several limitations. The primary constraint is the dependency on the quantity and variety of training data available. While the model performs well on enhanced datasets, its performance is inherently tied to the diversity and representativeness of the training data. Additionally, the computational overhead introduced by Sparse Coding and TV Regularization could limit the deployment of this model in low-resource environments or in real-time applications.

C. Future Research Directions

Future research should focus on several key areas to further enhance the utility and applicability of this model. Firstly, exploring more efficient implementations of Sparse Coding and TV Regularization could mitigate the computational costs and facilitate real-time processing. Additionally, expanding the dataset to include a wider array of gestures and more varied environmental conditions would likely improve the robustness and adaptability of the model. Moreover, integrating this model with other sensory data, such as depth information or temporal sequences, could provide a more holistic approach to gesture recognition.

D. Conclusion

In conclusion, this study underscores the efficacy of incorporating advanced regularization techniques into deep learning models for gesture recognition. The enhanced performance in noisy environments, coupled with improved generalization capabilities, offers promising avenues for both academic research and practical applications in the field of human-computer interaction.

REFERENCES

[1] V. Papyan, Y. Romano, and M. Elad, "Convolutional Neural Networks Analyzed via Convolutional Sparse Coding," arXiv:1607.08194, 2016.

[2] C.-H. Yeh et al., "Total Variation Optimization Layers for Computer Vision," CVPR, 2020.

[3] R. Nowak, "Deep Learning Meets Sparse Regularization: A Signal Processing Perspective," YouTube Lecture, 2020.

[4] J. Liu et al., "Image Restoration Using Total Variation Regularized Deep Image Prior," arXiv:2104.07016, 2021.

[5] J. He et al., "Make ℓ_1 Regularization Effective in Training Sparse CNN," arXiv:2006.08258, 2020.

[6] K. Gregor and Y. LeCun, "Learning Fast Approximations of Sparse Coding," in *Proc. ICML*, 2010.

[7] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[8] M. Han et al., "ESE: Efficient Speech Recognition Engine with Sparse LSTM on FPGA," in *Proc. FPGA*, 2017.

[9] J. Zhang, H. Li, and Y. Zhao, "TVConv: Total-Variation Constrained Convolutions for Robust Vision," in *Proc. CVPR*, 2023.

[10] Wikipedia contributors, "Fingerspelling — Wikipedia, The Free Encyclopedia," 2023. [Online]. Available: https://en.wikipedia.org/wiki/Fingerspelling. [Accessed: 10-April-2023].

[11] University of Toronto, Department of Mechanical & Industrial Engineering. “Assignment 2: Hand Gesture Recognition for MIE1517.” Course assignment, 2025.
[12] Dan Rasband, “ASL Alphabet Test,” 2016. [Online]. Available: https://www.kaggle.com/datasets/danrasband/asl-alphabet-test. [Accessed: 10-April-2023].

VII. Appendix

The code used in this project is available at the following URL: Google Colab Notebook

And also have been put on Github with corresponding results and model parapter files (.pt) at following URL: GitHub