CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset

Niraj Band
G H Raisoni College of Engineering
Nagpur,Maharashtra,India
Niraj.band.it.@ghrce.raisoni.net

Tejaswini Badpaiya
G H Raisoni College of Engineering
Nagpur,Maharashtra,India
tejaswini.badpaiya.it.@ghrce.raisoni.net

Swikruti thantharate
G H Raisoni College Of Engineering
Nagpur,Maharashtra, India
swikruti.thantharate.ds@ghrce.raisoni.net

Riya Darda
G H Raisoni College of Engineering
Nagpur,Maharashtra,India
riya.darda.it.@ghrce.raisoni.net

Sakshi Kadu
G H Raisoni College of Engineering
Nagpur,Maharashtra,India
sakshi.kadu.it.@ghrce.raisoni.net

Sania Sheikh
G H Raisoni College of Engineering
Nagpur,Maharashtra,India
sania.sheikh.it.@ghrce.raisoni.net

Mentor –
Mr. Anirvainya Gururajan
International Institute of Information Technology,
Hyderabad,Telangana,India

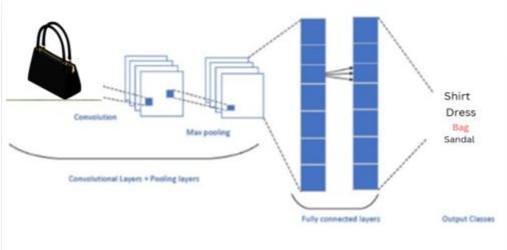
Abstract: Fashion items present a challenge in classification due to the wide array of styles, textures, and patterns. Convolutional neural networks (CNNs) are particularly effective for image classification, and this study proposes an enhanced CNN architecture tailored for fashion item categorization. By incorporating image augmentation and batch normalization, this model aims to boost performance and generalizability. Techniques like rotation, shifting, zooming, and flipping were applied to the images to strengthen model robustness. Additionally, a Batch Normalization layer was integrated mid-network to stabilize learning and expedite convergence. Training the model on an augmented dataset led to a test accuracy of 92.74%, a notable increase over a standard CNN model's 88.5% accuracy. Results suggest that combining image augmentation with Batch Normalization enhances CNN performance, making it more effective for fashion classification tasks.

INTRODUCTION

Fashion classification is a valuable task in computer vision, with applications in e-commerce, retail, and trend analysis, where it helps streamline operations and improve customer insights. However, fashion classification is challenging due to the vast variety in styles, patterns, and textures. Convolutional Neural Networks (CNNs) have shown strong performance in image classification by learning spatial features from visual data. Yet, CNN effectiveness often depends on architecture design and training methods, leading to continual research into optimization techniques.

This study introduces an enhanced CNN model for fashion image classification, emphasizing image augmentation and Batch Normalization. Image augmentation expands the dataset through transformations like rotations, shifts, zooms, and flips, which help the model generalize by exposing it to diverse data variations. Batch Normalization stabilizes the learning process by normalizing layer inputs, speeding convergence and improving model accuracy, making it a common feature in advanced deep learning models.

The proposed CNN architecture uses a combination of convolutional layers, Batch Normalization, Max Pooling, and fully connected layers, ending in a SoftMax layer for classification. Dropout layers also aid in regularization to prevent overfitting, and the model is optimized with the adaptive Adam optimizer. Tested on the Fashion-MNIST dataset, this improved CNN model demonstrated higher accuracy than a standard CNN, showcasing the effectiveness of image augmentation and Batch Normalization for robust and accurate fashion classification.



LITERATURE REVIEW

1. Dynamic CNN Models For Fashion Recommendation in Instagram : This paper describes the authors' research on two dynamic convolutional neural network models — "DynamicPruning" and "DynamicLayers" — for fashion recommendations based on Instagram images. The very models utilize post texts to increase performance of clothes and brands classification. As experimental results of the presented fashion CNN-based models pruned on Instagram and DeepFashion datasets, there was demonstrated a 20% rise in accuracy concerning base models with "DynamicLayers" giving up to 35% for multi-label classification. These approaches promote social media-driven recommendations through a more efficient fashion content filtration.
2. Enhanced Convolutional Neural Network for Fashion Classification: The research enhances the CNN architecture's performance in classifying fashion images through the incorporation of image augmentation as well as batch normalization techniques. With the use of Fashion-MNIST data, the enhanced CNN improved accuracy to 92.74%, which is a better result than that of a baseline model. Furthermore, more effective rotation and normalization techniques were able to improve generalization and stability leading to a better model performance on the various fashion images.
3. Fashion - MNIST Classification using CNN: This letter is written to demonstrate how Convolutional Neural Networks (CNNs) can be employed in classification of various articles of clothing using the fashion MNIST dataset containing 70,000 clothing fashion image samples. The CNN models assist in image classification through the application of techniques that increase the speed of training. The study demonstrates that CNNs could be effective in fashion image classification and may be relevant for virtual shopping and stock control.
4. Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture: The paper makes use of a CNN model called LeNet-5 for identifying clothing items in the Fashion-MNIST dataset (the dataset contains 70,000 images). The model managed to achieve more than 98% accuracy level which is better than the other models. This research demonstrates CNN's powerful capability of distinguishing among various clothing items, and it recommends testing such model on different fashion datasets in the future.
5. Classification of Fashion Article Images using Convolutional Neural Networks: The paper proposes a model that employs convolutional neural networks (CNNs) to identify the fashion images in the Fashion-MNIST dataset. The authors evaluate three architectures of CNN networks since the application of residual skip connections and batch normalization can facilitate learning. The presented models considerably outperformed other works with a margin of 2%, with the most accurate model attaining ***92.74%*** accuracy.
6. CNN Model for Images Classification on MNIST and Fashion-MNIST Dataset: This paper focuses on the classification of images using convolutional neural networks (CNN), and some tests were carried out on two datasets: MNIST and Fashion-MNIST. Five different designs of CNN models has been tried and some settings like optimizers and activation functions were modified. On all models, MNIST was performed well, but the third model evidenced the highest performance on significantly more difficult Fashion-MNIST data.

METHODOLOGY

Data Collection:For this study, the Fashion MNIST dataset from Zalando Research was used. This dataset contains 70,000 grayscale images, each measuring 28x28 pixels, representing 10 different categories like t-shirts, trousers, and bags. With 60,000 images for training and 10,000 for testing, the dataset is well-balanced, ensuring reliable performance evaluation. Before feeding the data into the model, we normalized the pixel values to a range of 0 to 1 by dividing them by 255, which helped the model train faster and made the computations more efficient.

This image is a sample dataset used in Convolutional Neural Network (CNN) image classification research. It appears to be similar to the Fashion-MNIST dataset, which contains grayscale images of various fashion items like shirts, shoes, and accessories, each labeled with their corresponding class (e.g., "Sneaker," "Ankle boot," "T-shirt/top," etc.). This dataset is commonly used for benchmarking image classification models, especially CNNs, due to its simplicity and high variance between classes.

Model Architecture:To classify these images, we opted for a Convolutional Neural Network (CNN) because CNNs are highly effective in image recognition tasks. The CNN architecture was designed to extract useful features, such as edges and textures, through several convolutional layers. To reduce the complexity of the data, pooling layers were used after each convolutional layer. To prevent the model from memorizing the training data, dropout layers were included, which randomly deactivate neurons during training, helping the model generalize better. In the final stages, dense layers were used to classify the images into their respective categories.

Layer Type	Description	Output Shape
Input Layer	28x28 grayscale image	(28, 28, 1)
Conv2D (64 filters)	Convolution layer with ReLU activation (3x3 kernel)	(26, 26, 64)
Conv2D (64 filters)	Second convolution layer with ReLU activation (3x3 kernel)	(24, 24, 64)
MaxPooling2D	Reduces spatial dimensions (2x2 pool size)	(12, 12, 64)
Dropout (0.25)	Randomly drops 25% of units to prevent overfitting	(12, 12, 64)
Conv2D (128 filters)	Third convolution layer with ReLU activation (3x3 kernel)	(10, 10, 128)
Conv2D (128 filters)	Fourth convolution layer with ReLU activation (3x3 kernel)	(8, 8, 128)
MaxPooling2D	Reduces spatial dimensions (2x2 pool size)	(4, 4, 128)
Dropout (0.25)	Randomly drops 25% of units	(4, 4, 128)
Flatten	Converts 2D feature maps to 1D vector	(2048,)
Dense (256 units)	Fully connected layer with ReLU activation	(256,)
Dropout (0.5)	Drops 50% of units	(256,)
Dense (10 units)	Output layer with Softmax activation (10 classes)	(10,)

The convolutional neural network (CNN) architecture described in Table 1 is designed to classify 28x28 grayscale images into 10 distinct classes. This model is suitable for datasets like Fashion-MNIST, focusing on efficient feature extraction and classification accuracy. The architecture comprises convolutional, pooling, dropout, and fully connected layers, structured as follows:

***Input Layer**: The input layer accepts 28x28 grayscale images with a single channel, providing a foundational input size for subsequent layers.*

***Convolutional Layers**: Two pairs of convolutional layers, each with 3x3 kernels and ReLU activations, progressively learn spatial hierarchies. The first pair consists of 64 filters, while the second pair uses 128 filters, increasing the network's capacity to capture complex features as depth increases. This design enhances feature extraction by detecting edges, textures, and shapes at multiple scales.*

***MaxPooling Layers**: Each convolutional block is followed by a MaxPooling layer with a 2x2 pool size to reduce spatial dimensions, minimizing computational requirements while retaining key features.*

***Dropout Layers**: To prevent overfitting, dropout layers with rates of 25% and 50% are applied after the convolutional and dense layers, respectively. Dropout randomly inactivates neurons during training, encouraging the network to generalize better on unseen data.*

***Flatten Layer**: After the final convolutional block, a flatten layer converts the 2D feature maps into a 1D vector. This transformation enables integration with fully connected layers for classification.*

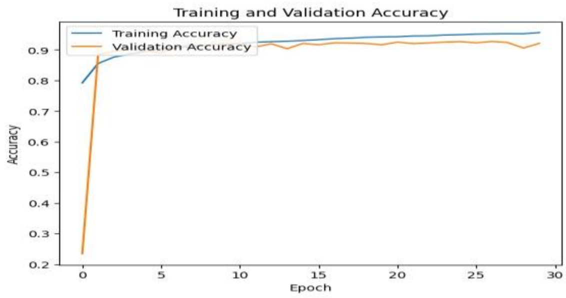
***Dense Layers**: The model includes two dense (fully connected) layers with 256 units and ReLU activation, providing high-capacity feature combination before classification. The final dense layer, with Softmax activation, outputs a probability distribution across the 10 classes, facilitating multiclass classification.*

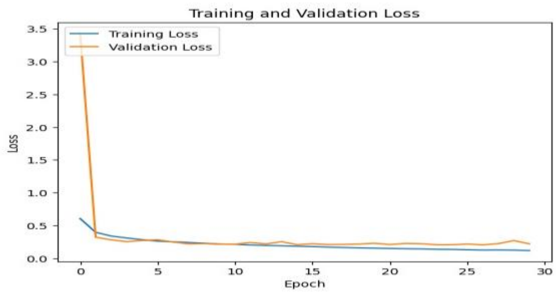
This architecture is optimized to balance computational efficiency and classification accuracy, leveraging convolutional layers for feature extraction, pooling for dimensionality reduction, dropout for regularization, and fully connected layers for decision-making.

Data Preprocessing: Before training the model, the dataset was split into training, validation, and test sets. The images were preprocessed by normalizing the pixel values, and no additional data augmentationn techniques were applied. A portion of the training set was used for validation to monitor the model's performance and adjust hyperparameters where necessary.

Training Process and Hyperparameters:The model was trained using the Adam optimizer, known for its ability to adapt the learning rate and converge more quickly than traditional gradient descent. We set the learning rate at 0.001, used a batch size of 64, and trained the model for 20 epochs. This balance ensured that the model had enough time to learn without overfitting. These specific hyperparameters were chosen based on testing to achieve the best accuracy.

	Precision	recall	F1-score	support
Class 0	0.88	0.89	0.88	1000
Class 1	0.99	0.99	0.99	1000
Class 2	0.93	0.82	0.87	1000
Class 3	0.93	0.94	0.94	1000
Class 4	0.84	0.93	0.88	1000
Class 5	1.00	0.98	0.99	1000
Class 6	0.79	0.78	0.79	1000
Class 7	0.96	0.98	0.97	1000
Class 8	0.99	0.99	0.99	1000
Class 9	0.98	0.97	0.97	1000





In above graph,Training and Validation Accuracy for Fashion-MNIST dataset Training and Validation Loss for Fashion-MNIST dataset

Data Analysis Techniques:Accuracy was our primary metric for evaluating the model on both training and test sets. However, we also generated a classification report that included precision, recall, and F1-score for each class, giving us a deeper understanding of how well the model was performing across all categories. Throughout the training process, we closely monitored both training and validation losses to catch any signs of overfitting or underfitting. If there were significant differences between these losses, adjustments were made.

To further improve the model's ability to generalize, we used dropout layers, and the learning rate of the Adam optimizer was carefully fine-tuned to ensure smooth and consistent training progress. These regularization techniques helped the model perform well on the test set.

Model Evaluation and Visual Analysis:To get a clearer picture of the model's learning behavior, we plotted the loss and accuracy curves for both the training and validation sets over the course of the training epochs. These visualizations made it easier to spot trends of overfitting or underfitting. In addition, we reviewed misclassified images to identify any recurring patterns or particularly difficult classes, providing insights for potential improvements.

CONCLUSION

This study presents a CNN model, optimized with data augmentation and Batch Normalization, to enhance classification performance on the Fashion-MNIST dataset. Data preprocessing included normalizing pixel values and reshaping images, while training leveraged augmentation techniques such as rotation, zoom, and shift to boost data diversity and generalization. The model architecture consists of two convolutional layers with 64 and 128 filters, Batch Normalization layers, MaxPooling, and Dropout to prevent overfitting. Using the Adam optimizer with a learning rate of 0.0005, the model was trained for 30 epochs, achieving a notable improvement in accuracy from 90% to 92.74% Visualization of sample predictions further confirmed the model's robust performance. These results underscore the effectiveness of careful architectural design, regularization, and data augmentation in achieving high accuracy for complex image classification tasks.

FUTURE SCOPE

Future work could focus on a few key directions:

1. Advanced Data Augmentation: Incorporating techniques like elastic distortions, cutout, and mixup could make models more robust by providing diverse data variations.

2. Transfer Learning: Using pretrained models can enhance model generalization, especially on smaller datasets, making CNNs applicable to a broader range of tasks.

3. Regularization and Optimization: Testing different regularization methods (e.g., weight decay) and optimizing hyperparameters could help reduce overfitting and improve performance.

4. Application to Complex Datasets: Extending these techniques to larger datasets like CIFAR-10 and ImageNet would show if they scale well in real-world settings.

5. Edge and Real-Time Applications: Optimizing CNNs for deployment on edge devices can reduce memory and latency, making them suitable for mobile and real-time applications.

These directions could further enhance the power and reliability of CNNs across diverse applications.

augmentation in achieving high accuracy for complex image classification tasks.

REFRENCES

1. https://www.researchgate.net/publication/384828851_Enhanced_Convolutional_Neural_Network_for_Fashion_Classification
2. https://sci-hub.se/10.1109/BDCloud.2018.00169
3. https://bhu.ac.in/research_pub/jsr/Volumes/JSR_64_02_2020/51.pdf
4. https://sci-hub.se/10.1109/ICIIP.2017.8313740
5. https://scihub.se/10.1109/ITCE48509.2020.9 047776
6. https://www.irjmets.com/uploadedfiles/paper//issue_7_july_2023/42990/final/fin_irjmets1692082330.pdf