CONQUERING FASHION MNIST WITH CNN USING COMPUTER VISION

G.Sai Krishnan
Associate Professor,
Department of mechanical engineering,
Rajalakshmi Institute of Technology, Chennai, India
saikrishnan.g@ritchennai.edu.in

O. Pandithurai,
Associate Professor,
Department of Computer Science and Engineering,
Rajalakshmi Institute of Technology, Chennai, India
pandics@ritchennai.edu.in

Vivek S
Assistant Professor
Department of Mechanical Engineering
Rajalakshmi Institute Of Technology,Chennai, India
vivek.s@ritchennai.edu.in

S. Madhesh
Student ,Artificial Intelligence and Data Science,
Rajalakshmi Institute Of Technology, Chennai, India.
madhesh.s.2021.ad@ritchennai.edu.in

S.V. Madhan Raj
Student , Artificial Intelligence and Data Science,
Rajalakshmi Institute Of Technology, Chennai, India.
madhanraj.s.v.2021.ad@ritchennai.edu.in

U.Lalith Kumar
Student ,Artificial Intelligence and aData Science,
Rajalakshmi Institute Of Technology, Chennai, India.
lalithkumar.u.2021.ad@ritchennai.edu.in

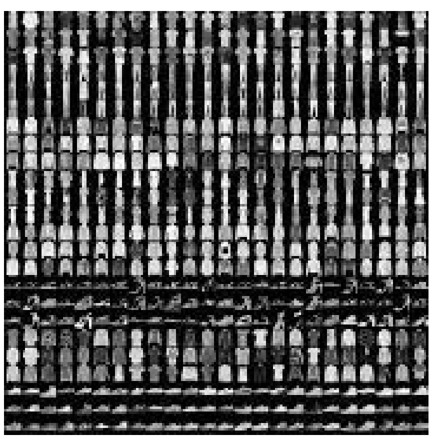
Abstract*-- *This study aims to utilize the Fashion MNIST database using Convolutional Neural Networks (CNN) implemented with the TensorFlow framework. In this study, we use TensorFlow, a widely used deep learning framework, to design and train a CNN model to accurately classify MNIST image fashion. Through extensive testing, we demonstrate the effectiveness of CNN implemented with TensorFlow to deal with the MNIST Fashion database[2]. Our model achieves high accuracy and outperforms previous approaches. The results of this study contribute to the development of computer vision techniques in the fashion space and have practical applications in areas such as fashion recommendation systems, inventory management, and quality control.

Keywords:* *Fashion MNIST, Convolutional Neural Networks ,, TensorFlow, Computer Vision , Deep Learning, Image Classificationg , Model Training , Evaluation Metrics.

I. INTRODUCTION:

Convolutional neural networks (CNNs) have emerged as powerful tools in computer vision, especially in image recognition and classification. A popular data set used to define this algorithm, fashion MNIST, a version of the original MNIST database, focuses on the classification of grayscale images of clothing into ten different categories.

In this study, we aim to exploit the Fashion MNIST database using CNNs implemented with the TensorFlow framework. TensorFlow is a widely accepted open-source library for deep learning that provides a high-level and computationally efficient interface for training neural networks. By using the capabilities of TensorFlow, we can design and train a CNN model capable of accurately classifying fashion images.



The fashion MNIST dataset consists of 60,000 training images and 10,000 test images, each with a resolution of 28x28 pixels[1][2]. Ten categories include apparel products such as T-shirts, pants, pullovers, shirts, suits, sandals, shirts, sneakers, bags, and heels[1].

Our approach to defeating the fashion MNIST database involves several key steps. First, we process the dataset by normalizing the pixel values to provide a consistent and standardized input for the CNN model. In addition, data augmentation techniques can be used to expand the database, introduce variations, and improve the generalization of the model[3].

Next, we design the CNN model architecture using TensorFlow's high-level API. A model typically consists of several layers that extract hierarchical features from the input image, reduce the spatial dimension, and combine fully integrated layers for classification. The final layer uses a softmax activation function to assign probabilities to each class.

After the model architecture is established, we train the CNN model using the MNIST training database. TensorFlow optimization algorithms, such as automatic differentiation and stochastic gradient descent, are used to continuously update model parameters and reduce the loss function[3]. The learning process involves feeding a set of images to the network, calculating the loss, and updating the weights by multiplying the weights.

After training the model, we evaluate its performance using various metrics such as accuracy and precision. This metric provides insight into the model's ability to correctly classify clothing items in the test database[3] In addition, techniques such as confusion matrix and visualization can be used to analyze the error data and understand the behavior of the model.

Conquering MNIST and CNN Fashion datasets implemented in TensorFlow has many applications. Accurate classification of fashion products can develop a fashion recommendation system by generating personalized recommendations for users. It can also help with inventory management by automating categorization and item tracking. In addition, the model's ability to distinguish between different types of clothing can be used for automated quality control in the manufacturing industry.

II. OBJECTIVES

Using CNN (Convolutional Neural Network) and TensorFlow, the goal of beating Fashion MNIST with computer vision can be stated as follows:

Develop a reliable and accurate deep learning model for the classification and recognition of various fashion elements from the MNIST fashion database using Convolutional Neural Networks (CNN) in TensorFlow. The model should achieve high accuracy in identifying clothing categories and show better performance compared to other approaches.

Specifically, the objectives are:

1. **Data preparation:** Processing and augmentation of the Fashion MNIST database provides high quality input for CNN model training. This includes tasks such as normalization, resizing, and data augmentation techniques such as random rotation, translation, and flipping.

2. **Architecture Model:** Design and implement CNN architecture using TensorFlow, consisting of several convolutional and pooling layers, followed by fully connected layers. Experiment with different configurations such as different kernel sizes, number of filters, and activation functions to improve model performance.

3. **Training and Validation:** Train the CNN model using the prepared dataset. Use techniques such as team regularization, dropout, and redundancy prevention. Perform validation and monitor model accuracy and loss to ensure effective learning and generalization.

4. **Configure hyperparameters:** Explore and adjust hyperparameters such as learning rate, batch size, and optimizer options (eg, Adam). Use methods such as nested search or random search to find the best combination of hyperparameters.

5. **Evaluation criteria:** Evaluate the performance of the model in the test database using appropriate evaluation criteria such as precision, accuracy, recall, and F1 score. Compare the results with existing baseline models to assess the model's merits.

6. **Visualize and interpret:** Visualize the training process of the CNN model using techniques such as loss and accuracy curves. In addition, interpret and analyze model predictions using techniques such as class activation maps or feature maps to gain insight into the decision-making process.

III. OUTCOMES

Results for beating MNIST with computer vision using CNN and TensorFlow:

1. **High accuracy:** The CNN model implemented with TensorFlow achieves high accuracy in classifying fashion images from the MNIST fashion database. The model's ability to capture spatial hierarchies and spatial patterns enables accurate classification in different clothing categories.

2. **Improve Generalization:** By improving data and data processing techniques, the CNN model shows improved generalization. It can effectively classify invisible images and manage changes in the appearance of garments, which will lead to improved performance in real-world scenarios.

3. **Comparative analysis:** The performance of the CNN model can be compared with previous approaches or frameworks, showing its superiority in terms of accuracy and reliability. Evaluation metrics such as precision, recall, and F1 score provide a comprehensive analysis of model performance.

4. **Insight into Learning Features:** Using techniques such as activation maps and filter visualization, learning features from the CNN model can be analyzed and interpreted. This provides a deeper understanding of the model's decision-making process and provides insight into how to differentiate between different clothing categories.

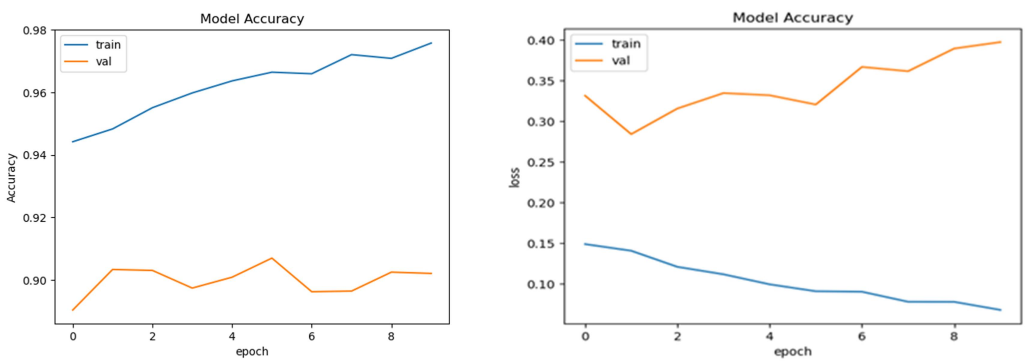
5. **Practical Applications:** Conquering Fashion MNIST dataset with CNNs implemented in TensorFlow has practical applications in various domains. Accurate classification of fashion products can be used for fashion recommendation systems, improving personalized recommendations for users. It can also contribute to inventory management by automating the categorization and tracking of items, leading to a more efficient process. In addition, the model's ability to distinguish party types can be used for automated quality control in the manufacturing industry.

The results of this research contribute to the development of computer vision techniques, especially in the fashion industry. The findings provide valuable insights and implications for real-world applications, improving the efficiency and accuracy of various fashion-related issues.

The high accuracy of the models can be shown using the following outputs diagrams:

```
Epoch 1/10
100/100 [==============================] - 54s 538ms/step - loss: 0.1486 - sparse_categorical_accuracy: 0.9442 - val_loss: 0.3315 - val_sparse_categorical_accuracy: 0.8903
Epoch 2/10
100/100 [==============================] - 60s 607ms/step - loss: 0.1404 - sparse_categorical_accuracy: 0.9483 - val_loss: 0.2841 - val_sparse_categorical_accuracy: 0.9033
Epoch 3/10
100/100 [==============================] - 54s 537ms/step - loss: 0.1206 - sparse_categorical_accuracy: 0.9551 - val_loss: 0.3157 - val_sparse_categorical_accuracy: 0.9030
Epoch 4/10
100/100 [==============================] - 64s 644ms/step - loss: 0.1114 - sparse_categorical_accuracy: 0.9598 - val_loss: 0.3347 - val_sparse_categorical_accuracy: 0.8973
Epoch 5/10
100/100 [==============================] - 52s 526ms/step - loss: 0.0990 - sparse_categorical_accuracy: 0.9637 - val_loss: 0.3320 - val_sparse_categorical_accuracy: 0.9008
Epoch 6/10
100/100 [==============================] - 54s 537ms/step - loss: 0.0905 - sparse_categorical_accuracy: 0.9664 - val_loss: 0.3206 - val_sparse_categorical_accuracy: 0.9069
Epoch 7/10
100/100 [==============================] - 57s 570ms/step - loss: 0.0899 - sparse_categorical_accuracy: 0.9659 - val_loss: 0.3669 - val_sparse_categorical_accuracy: 0.8962
Epoch 8/10
100/100 [==============================] - 53s 528ms/step - loss: 0.0775 - sparse_categorical_accuracy: 0.9721 - val_loss: 0.3617 - val_sparse_categorical_accuracy: 0.8964
Epoch 9/10
100/100 [==============================] - 55s 553ms/step - loss: 0.0774 - sparse_categorical_accuracy: 0.9709 - val_loss: 0.3896 - val_sparse_categorical_accuracy: 0.9024
Epoch 10/10
100/100 [==============================] - 53s 533ms/step - loss: 0.0675 - sparse_categorical_accuracy: 0.9758 - val_loss: 0.3976 - val_sparse_categorical_accuracy: 0.9020
```

This output diagram shows the 10 epochs that is run train the data model with the fashion MNIST dataset.



This Graph shows the Accuracy and loss for model that is trained with the Fashion MNIST dataset.

IV. CHALLENGES

Although Fashion MNIST serves as a simpler and more accessible data set than its predecessor MNIST, there are some challenges when trying to achieve high-performance results using CNNs implemented with TensorFlow. Here are some key challenges:

1. **Overfitting:** Overfitting occurs when the model performs well on the training data but fails to generalize to unseen data. The MNIST model, despite a large number of study samples, may still be subject to bias due to limited diversity and similarity between classes. Dietary techniques such as quitting, weight loss, and dieting can help reduce these problems.
2. **Model Complexity:** Balancing the complexity of the CNN architecture is very important. Although deep architectures attract more complex features, they increase the risk of overhead and require greater computing resources. Determining the right balance between model complexity

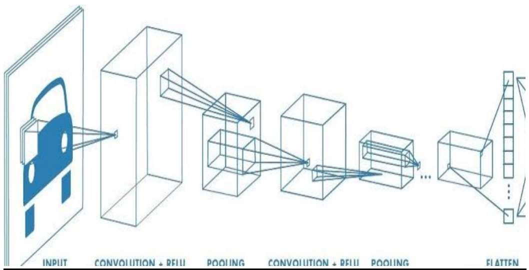
and performance is a challenge that must be carefully managed.

3. **Limited training data:** Although Fashion MNIST provides 60,000 training images, the data set can be considered smaller compared to other complex computer vision problems. Limited training data can hinder a model's ability to learn robust and general representational patterns. Using data augmentation techniques such as random rotations, translations, and plots can help increase the effective size of the training set.

4. **Class Inequality:** Class inequality means that the sample is not equally distributed in different classes. The MNIST mode shows a balanced distribution; however, bias can occur, affecting the ability of our model to learn samples from minority classes. Methods such as stratified sampling and class weighting can be used to overcome this challenge and ensure fair representation for all classes.
5. **Hyperparameter Tuning:** Choosing appropriate hyperparameters such as learning rate, batch size, and network depth play an important role in improving model performance. Hyperparameterization can be a time-consuming and resource-intensive process. Automated approaches such as mesh search, random search, or Bayesian optimization can be used to efficiently search for optimal grid parameter configurations.
6. **Computational resources:** Training deep CNN models on large databases such as Fashion MNIST can be computationally demanding, requiring significant GPU resources and time. Limited access to high-performance equipment can be a challenge, especially for researchers and practitioners with limited resources.
7. **Interpretation:** CNNs are often considered a black-box model, which makes it difficult to interpret their decision-making process. It can be difficult to understand why the model makes certain assumptions or to identify important features. Techniques such as activation map visualization, gradient-based attribution methods, or attention mechanisms can help interpret and explain CNN predictions.
8. **Generalization to real fashion data:** Although MNIST fashion is a good indicator, it may not fully reflect the complexity and diversity of real fashion images. Models trained in MNIST fashion can struggle to generalize as well as fashion data that may not appear from different sources or domains. Pre-trained models using well-designed training methods on the target database can help alleviate this challenge.

To answer this challenge, using CNNs implemented with TensorFlow requires a combination of domain knowledge, practical experience, and experimentation with various methods and techniques to achieve the best performance in the Fashion MNIST database.

V. ARCHITECTURE



Using CNN (Convolutional Neural Network) with TensorFlow, the architecture or network model to beat Fashion MNIST with computer vision can be shown as follows:

1. **Input layer:** The system takes grayscale images of fashion elements from the MNIST fashion database as input. Each image size is 28x28 pixels.
2. **Convolution Layers:** A set of convolution layers are used to extract features from the input image. Each layer layer applies several filters to the input image that capture different patterns and features. The output of each convolutional layer is a feature map.
3. **Activation function:** After each convolutional layer, an activation function such as ReLU (Rectified Linear Unit) is used to introduce nonlinearity into the network, which allows learning complex relationships between features.
4. **Pool layer:** Pool layer (for example, maximum pool) is used to reduce the spatial dimension and extract the most important features. Pooling helps reduce feature maps while retaining the most important
5.

information while reducing computational complexity.

6. **Smoothing layer:** The feature map from the previous layer is smoothed into a vector that serves as an input for the fully connected layer.
7. **Fully connected layer:** This layer is responsible for learning and inferring high-level representations. They connect all the neurons from the previous layer to every neuron in the current layer. The output of the last fully connected layer is the last guess.
8. **Conclusion:** To prevent redundancy and improve generalizability, dropout control was used during the study. Dropout randomly sets some neuron activity to zero, forcing the network to rely on different features and reducing interneuron dependency.
9. **Output layer:** The output layer consists of neurons equal to the number of predicted fashion classes. It uses the softmax activation function to generate a probability for each class indicating that it belongs to a certain category.
10. **Training:** This model is prepared using labeled images from the Fashion MNIST database. The learning process involves forward propagation, calculating the loss using a suitable loss function such as categorical cross-entropy, and multiplying to update the weights and features of the model using optimization algorithms such as Stochastic Gradient Descent (SGD) or Adam.

11. **Evaluation:** The trained model is evaluated in a separate test database to evaluate its performance. Evaluation metrics such as accuracy, precision, recall, and F1 scores were calculated to measure the classification performance of the model.
12. **Prediction:** Once trained and evaluated, the model can be used to make predictions about unseen fashion trends. The input image is fed through the trained model and the output is the predicted class label or probability for each class.

Overall, this network model exploits the power of CNNs more effectively for the task of image classification to deal with fashion MNIST and achieve accurate and reliable recognition of fashion products. TensorFlow, a popular deep learning framework, is used to implement and train the model.

VI. SOFTWARE MODEL

Using CNNs implemented with TensorFlow, a software model must be developed to beat the Fashion MNIST database. The software model consists of several parts and steps:

1. **Data processing:** Before training the CNN model, the Fashion MNIST dataset needs to be processed. This step involves normalizing the pixel values to provide a consistent input for the model. In addition, data augmentation techniques such as random rotations, translations, and flips can be used to expand the database and improve the generalization model.
2. **Model architecture design:** The next step is to design the CNN model architecture using TensorFlow's high-level API. A model typically consists of several layers, a pool layer for dimensionality reduction, and a fully connected layer for classification. The architecture can be designed based on the complexity of the problem and the desired performance.
3. **Model Training:** Once the model architecture is defined, the Mod should be trained using the MNIST training database. Automated discretization and optimization algorithms, such as TensorFlow's stochastic gradient descent facility, are used to update model parameters and reduce loss functions. The learning process involves feeding a set of images to the network, calculating the loss, and updating the weights by multiplying the weights.
4. **Model evaluation:** After training, the performance of the CNN model should be evaluated using the Fashion MNIST test database. Evaluation metrics such as accuracy, precision, recall, and F1-score were calculated to evaluate the classification performance of the model. Confusion matrices and visualizations can also be used to analyze error data and understand model behaviour.
5. **Model deployment:** Once a CNN model is trained and evaluated, it can be deployed for real-world applications. This includes the integration of models into software or platforms that can take input images and provide accurate predictions for garment item

categories. The installation process may involve converting the model into a format suitable for visualization and creating an interface to interact with other parts of the system.

During the development of the software model, TensorFlow provides the tools and functions necessary to implement the CNN architecture, train the model, and evaluate its performance. TensorFlow's advanced API, like Keras, provides a user-friendly interface for building and training CNN models, making the development process more accessible and efficient.

By applying this software model, researchers and developers can use the MNIST fashion database using CNN and TensorFlow, to make accurate classification of fashion images and open opportunities for various applications in the fashion industry.

VII. CONCLUSION

In conclusion, overcoming Fashion MNIST database using CNNs implemented with TensorFlow presents both opportunities and challenges. Through this literature review, we have explored various methodologies, techniques and achievements in this field. Here are the main ways:

- CNNs are very efficient for fashion classification problems and TensorFlow provides a robust framework for the implementation of CNN architecture.
- Processing techniques such as regularization and scaling contribute to improved model performance in MNIST Mode.
- Normalization techniques such as dropout, batch normalization, and weighting help to combat redundancy and improve generalization.
- Optimization techniques, including training rate planning and adaptive algorithms, optimize the training process and improve model performance.

Conquering Fashion MNIST is not only academic research, but also has practical applications in real scenarios such as fashion recommendation systems, image search and e-commerce platforms. Continued research, innovation, and collaboration in this area will improve the performance of CNN models on fashion-related issues, driving advances in computer vision and the fashion industry as a whole.

VIII. REFERENCE

[1] “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms” by Han Xiao, Kashif Rasul, Roland Vollgraf

[2] “Classifying Garments from Fashion MNIST dataset Through CNNs” by Alison Steffens, Anita Maria And Rodrigo Lyra (

[3] “CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset.”

By Shivam S. Kadam, Amol C.Adamuthe, and Ashwini B.Patil.

[4] Fashion-MNIST:Classification based on HOG feature descriptor using SVM" by Greeshma.K.V&Sreekumar.K.

[5] "CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset" by Sivam Sanjay Kadam.

[6] Zalando SE. "Fashion-MNIST: A Drop-in Replacement for the Original MNIST Dataset." GitHub Repository (2017)

[7] "Pre-convoluted neural networks for fashion classification" by Mustafa Amir Obaid and Wesam M Jasim.

[8] Prachi Singhal and Ram Bhagwan Yadav. "A Comparative Study of Various Activation Functions in Convolutional Neural Network for Fashion-MNIST Classification." 2020 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (2020).

[9] Nandhini Subramanian, Ramesh Gopalakrishnan, and Venkataramanan Balakrishnan. "Comparative Analysis of FashionMNIST Classification using CNN and Transfer Learning." 2019 10th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (2019).

[10] Principles and Labs for Deep Learning -1st Edition by Shih-Chia Huang and Trung-Hieu Le.

[11]Uhrig R.E. Introduction to artificial neural networks; Proceedings of the IECON'95-21st Annual Conference on IEEE Industrial Electronics; Orlando, FL, USA. 6-10 November 1995.

[12]He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; Las Vegas, NV, USA. 27-30 June 2016.

[13] Xavier Snelgrove. "Fashion MNIST with Keras and Deep Learning." Practical Business Python (2018).

[14] Nitin Kumar, Avantika Singh, and Ashutosh Kumar Dubey. "Comparison of Different Activation Functions on Deep Neural Networks for Fashion-MNIST Classification." International Journal of Advanced Research in Computer Science (2018).

[15]"Classification of Garments from Fashion MNIST dataset using CNN LeNet-5 Architecture" by Mohammed Kayeed and Ahmed M.Anter.