Image Classification

Fashion MNIST

Objective

Develop machine learning model that can accurately classify the images into their respective fashion categories

Dataset

- Fashion MNIST is a dataset of Zalando's article images
- Fashion MNIST contains of a training set of ***60,000*** examples and a test set of ***10,000*** examples.
- Each example is a 28x28 grayscale image, associated with a label from ***10 classes***.
- The intensity of the image, ranging from 0 (black) to 255 (white).

Train & Test Sets Examples

Labels:

- 0 - T-shirt/top
- 1 - Trouser
- 2 - Pullover
- 3 - Dress
- 4 - Coat
- 5 - Sandal
- 6 - Shirt
- 7 - Sneaker
- 8 - Bag
- 9 - Ankle boot

Data Wrangling

- Each row is a separate image.
- Column 1 is the class label.
- Remaining columns are pixel numbers (784 total).
- Each value is the darkness of the pixel (1 to 255).

Data Pre-processing

1- Conversion to Categorical Metrics

- Labels are converted to one-hot encoded categorical metrics

2- Feature Extraction and Reshaping

- The pixel values (feature columns) are extracted and reshaped to the dimensions of the input images
- The feature array is converted into a 4-dimensional array, suitable for image data

3- Normalization

- The pixel values are normalized by dividing them by 255, bringing them within the range of 0 to 1

4- split training data into 80% training and 20% validation.

Methodology

- For this project we will be going to use deep learning concepts like artificial neural networks and convolutional neural networks to build an image classification model which will learn to distinguish 10 different item images into their respective categories.

Modelling

- For building an image classifier here we use deep convolutional neural networks (CNN).

Model Architecture:

- Convolution Layer: The primary purpose of Convolution in case of a CNN is to extract features from the input image.
- Max Pooling Layer: Pooling reduces the dimensionality of each feature map but retaining the most important information.
- Fully Connected Layer/Dense Layer: “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer.

Convolutional Network

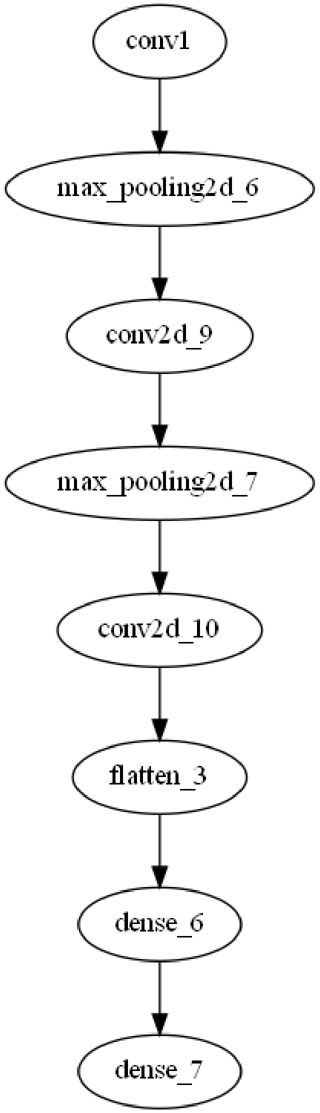
Training process in a Convolution Network:

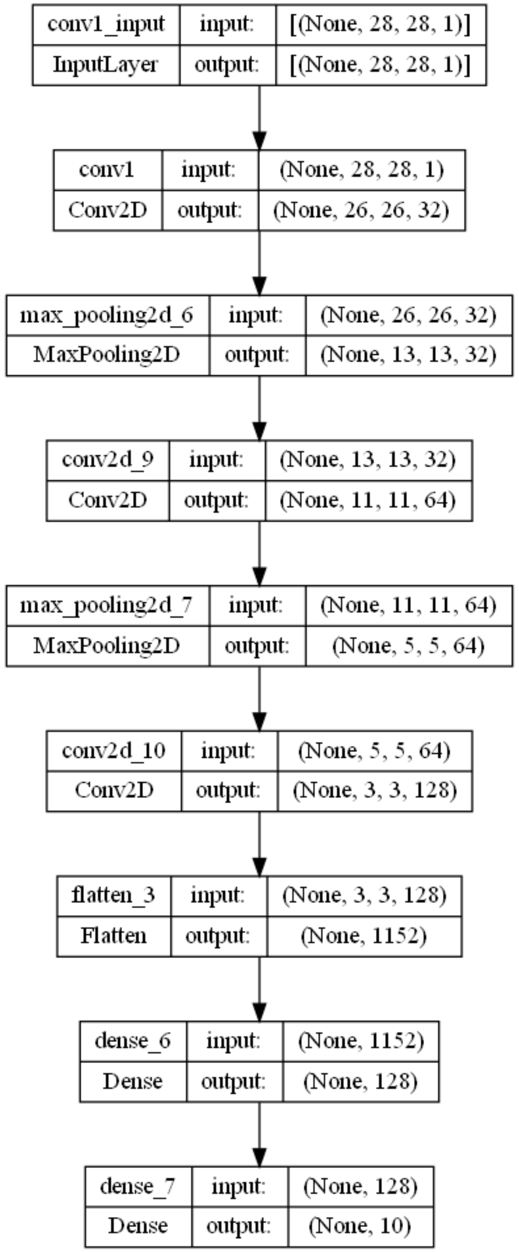
- We initialize all filters and parameters / weights with random values.
- The network takes a training image as input, goes through the forward propagation step (convolution, ReLU and pooling operations along with forward propagation in the Fully Connected layer) and finds the output probabilities for each class.
- Calculate the total error at the output layer (summation over all 10 classes).
- Use Backpropagation to calculate the gradients of the error with respect to all weights in the network and use gradient descent to update all filter values / weights and parameter values to minimize the output error.

Evaluation Metric

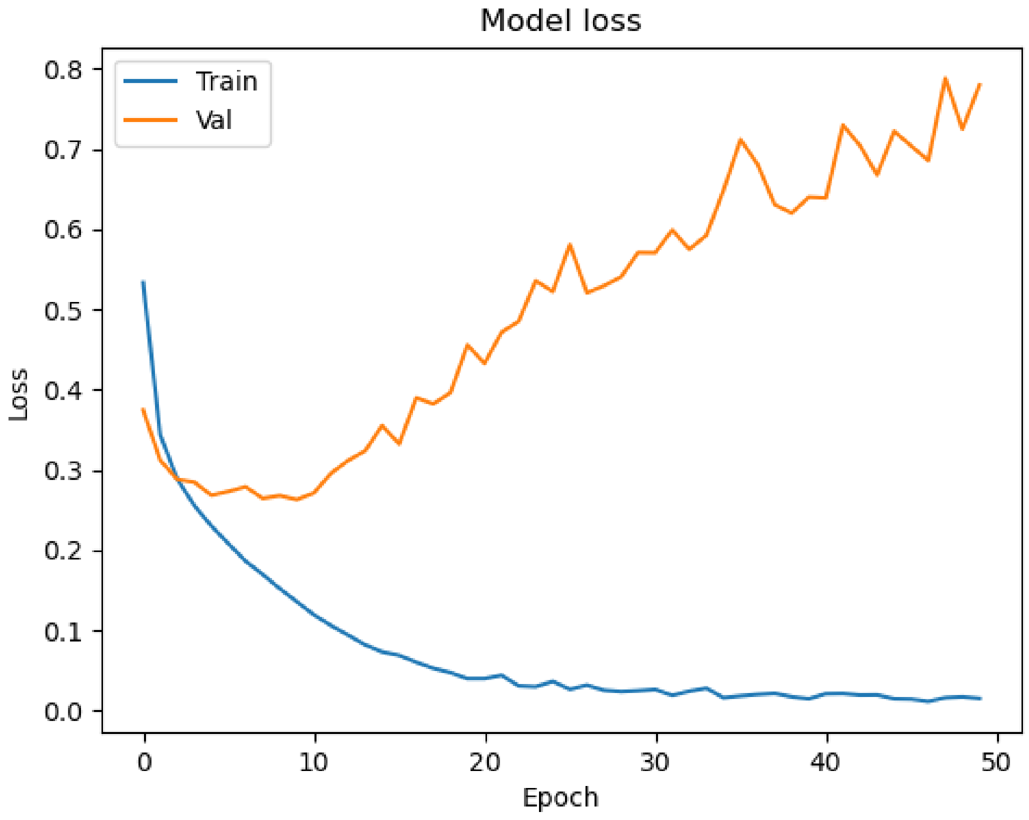
- Evaluation metrics explain the performance of a model.
- Categorical Cross-Entropy & Accuracy are our metrics in this project.
- **Categorical Cross-Entropy:** Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label.

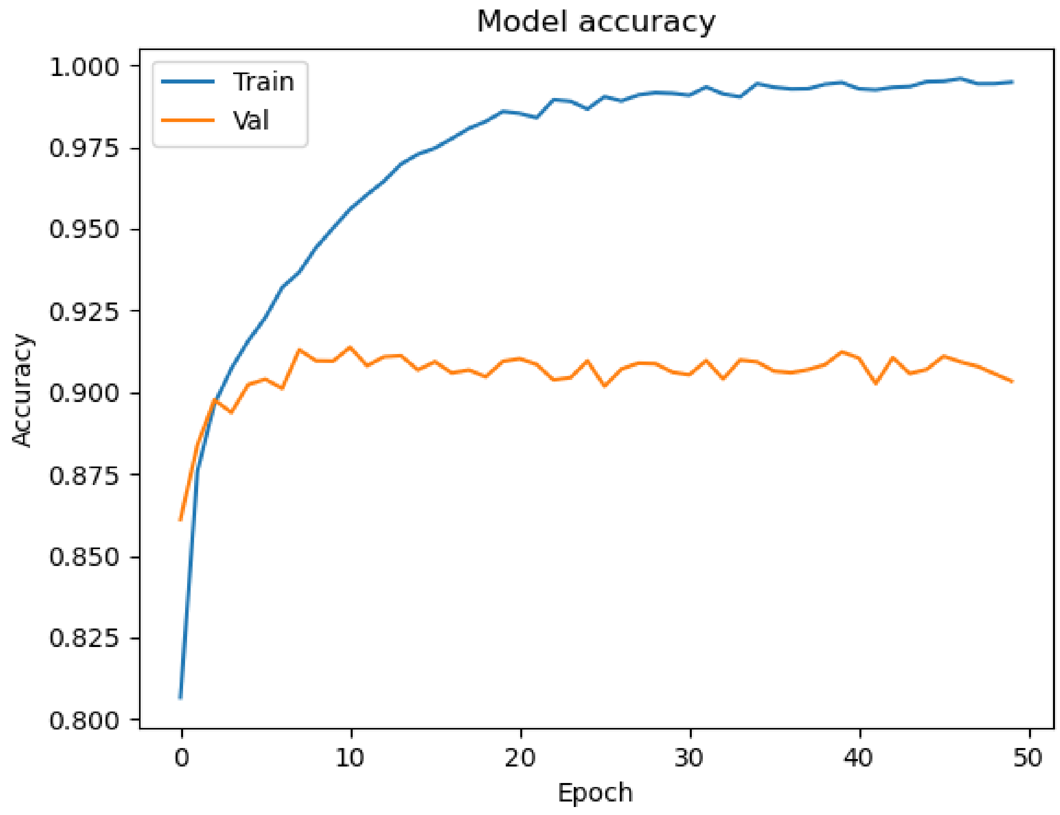
Model Architecture





Loss and Accuracy plots





Generalization & Regularization

- Generalization refers to how well the concepts learned by the model apply to new unseen data.
- Overfitting happens when the models learns too well the details and the noise from training data, but it doesn't generalize well, so the performance is poor for testing data.
- Regularization is a key component in preventing overfitting.

Regularization techniques:

1. Dropout.
2. Batch Normalization.

Hyper Parameters

- Hyper parameters are the variables which determines the network structure and the variables which determine how the network is trained.

Hyper parameters related to Network structure:

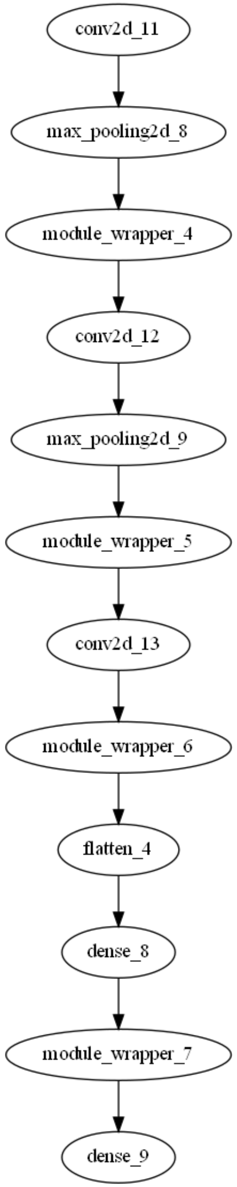
- Number of Hidden-Layers & Units
- Dropout

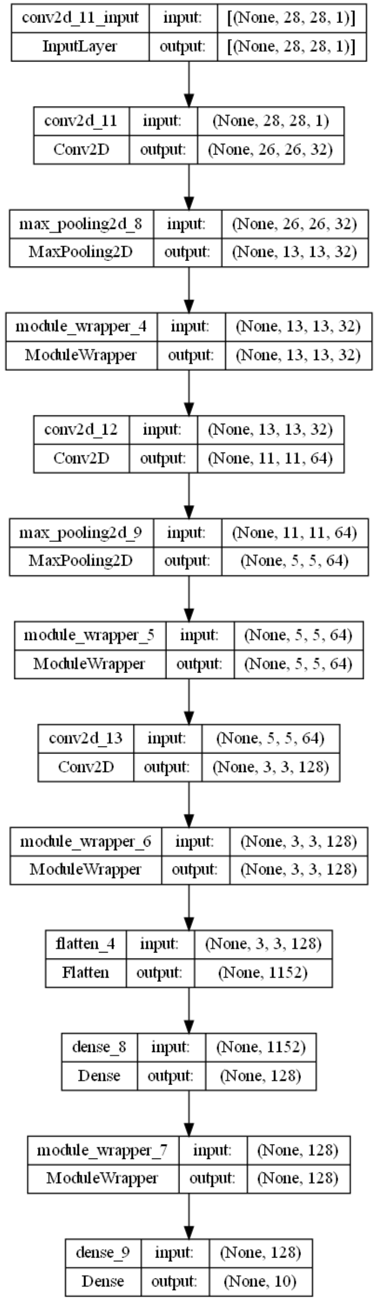
Hyper parameters related to Training Algorithm:

- Number of Epochs
- Batch size

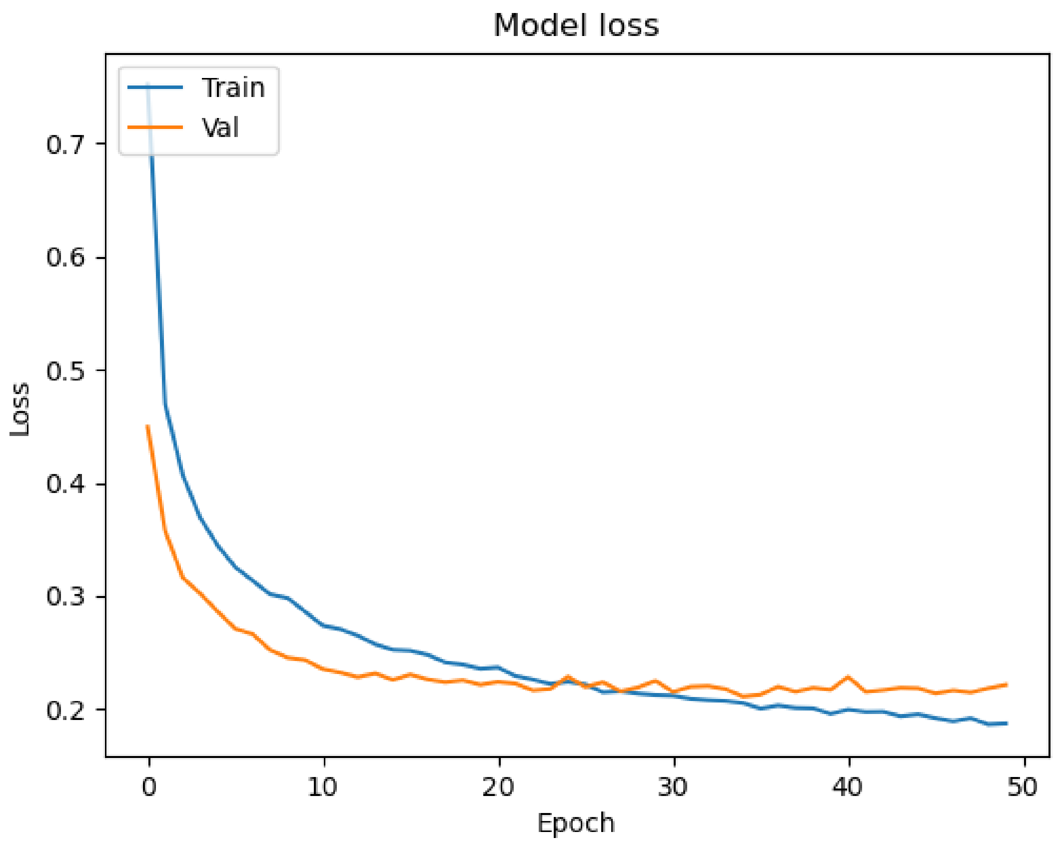
Modified Model Architecture

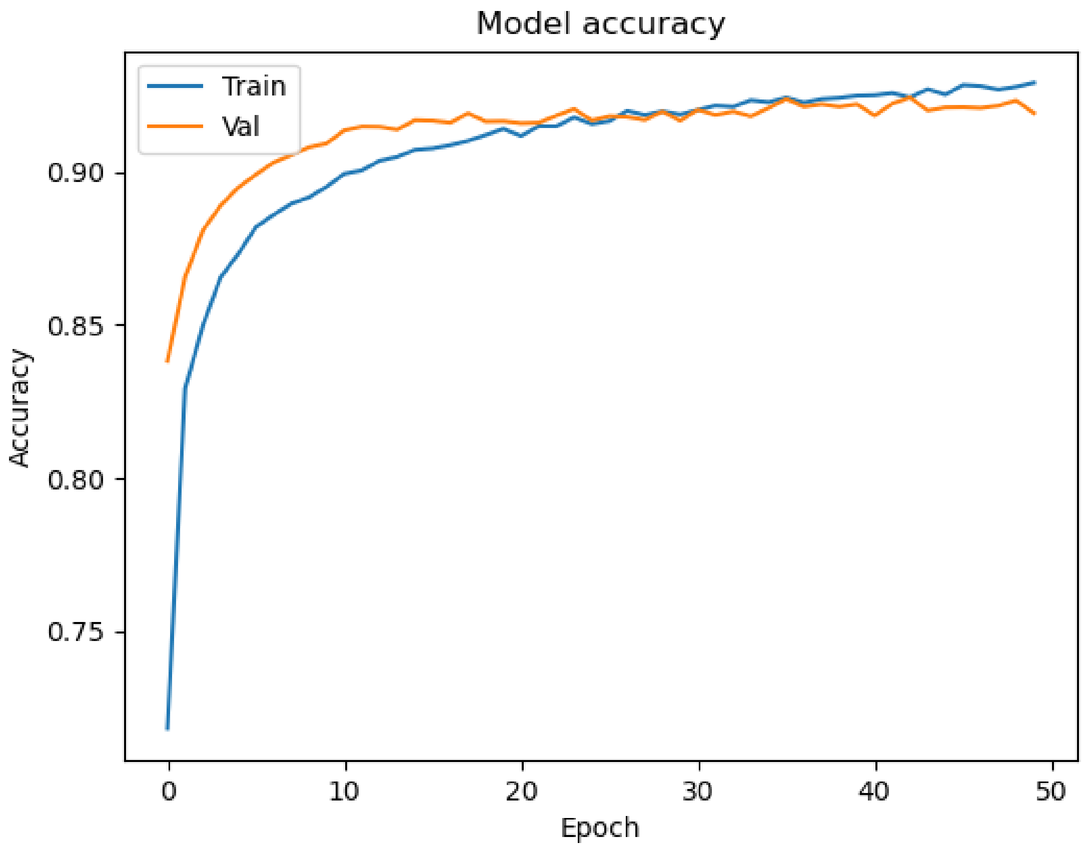
- This is how the final model is layered:
  - Three Convolution Layers with 32, 64, 128 filters with a 3x3 kernel and padding.
  - Three Max Pooling Layer with 2x2 pooling size.
  - Four Dropouts layers with 0.25, 0.25, 0.4 and 0.3 rates.
  - Flatten Layer.
  - Two Fully Connected Layer with 128 and 10 neurons.
  - Output Layer with 10 neurons.





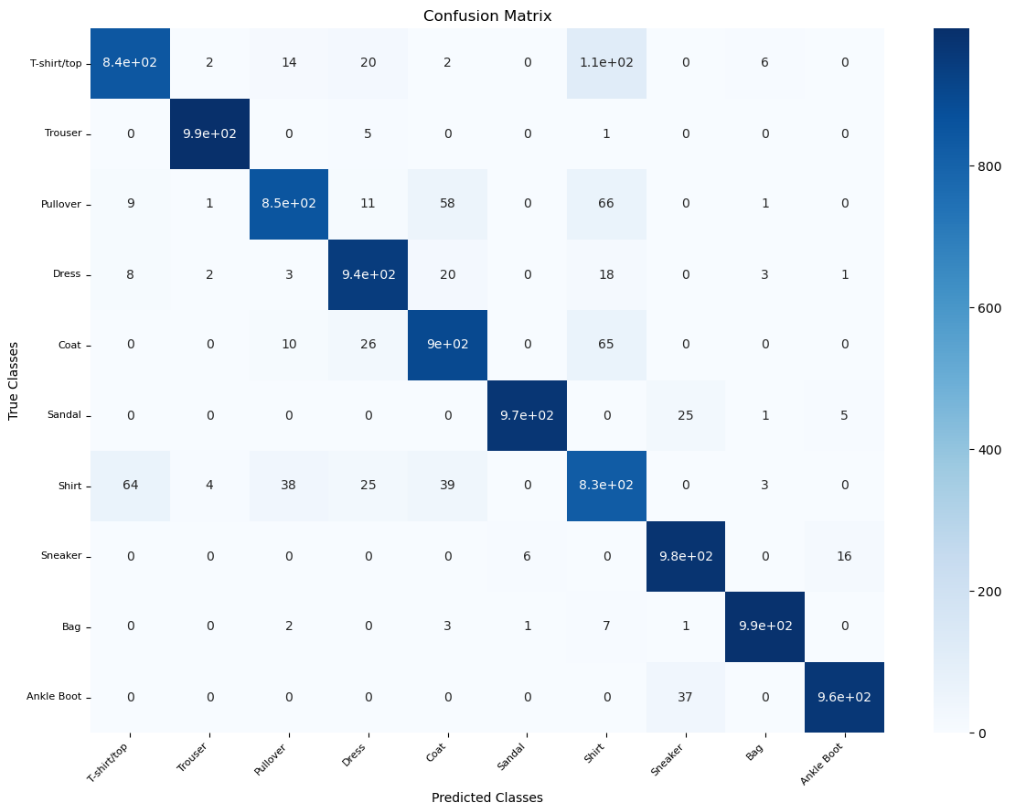
Loss and Accuracy plots





Test Set Results

- Test Loss: 0.2088
- Test Accuracy: 0.9258



	precision	recall	f1-score	support
Class 0 (T-shirt/top) :	0.91	0.84	0.88	1000
Class 1 (Trouser) :	0.99	0.99	0.99	1000
Class 2 (Pullover) :	0.93	0.85	0.89	1000
Class 3 (Dress) :	0.92	0.94	0.93	1000
Class 4 (Coat) :	0.88	0.90	0.89	1000
Class 5 (Sandal) :	0.99	0.97	0.98	1000
Class 6 (Shirt) :	0.75	0.83	0.79	1000
Class 7 (Sneaker) :	0.94	0.98	0.96	1000
Class 8 (Bag) :	0.99	0.99	0.99	1000
Class 9 (Ankle Boot) :	0.98	0.96	0.97	1000
accuracy			0.93	10000
macro avg	0.93	0.93	0.93	10000
weighted avg	0.93	0.93	0.93	10000

Misclassified Examples

Strength

- The model achieved a high accuracy of 92.58% on the test set, indicating its ability to classify fashion items accurately.
- The addition of dropout layers helped mitigate overfitting and improve generalization, resulting in better performance on the validation set.
- The model demonstrated good precision, recall, and F1-score across most classes, indicating its ability to correctly classify different fashion items.
- The confusion matrix provided insights into the performance of the model, highlighting areas where misclassifications occurred.

Limitations

- Despite achieving good overall accuracy, the model struggled with certain classes, such as shirts and pullovers, where precision and recall were relatively lower. This suggests that the model may have difficulty distinguishing between similar clothing items.
- The model's performance plateaued after reaching an accuracy of around 0.9, indicating a potential limitation in its ability to further improve accuracy on the validation set.
- The model may be sensitive to variations in image quality, lighting conditions, or other factors that could impact the visual appearance of fashion items.
- The dataset itself may have limitations, such as class imbalance or inconsistencies in labeling, which could affect the model's performance.

Business Impact

- Product Categorization and Recommendations
  - Putting images of new products
  - Personalized recommendations based on preferences and browsing history
- Quality Control and Defect Detection
  - Detect anomalies or deviation from the expected standards
- Trend Analysis and Market Insight
  - Inform decision-making processes, like designing new collections
  - Tailoring marketing campaigns to align with fashion trends