Project Title: **FASHION PRODUCT CLASSIFICATION USING DEEP LEARNING**

Team Name: **LEO**

TEAM MEMBERS:

1. Akshwin Kisore – 20BEC0521
2. Nandha Kishore – 20BEC0667
3. Jaisuryaa P – 20BEC0296
4. D.V.S Kousik – 20BEC0355
5. Akshavi Raj – 20BEC0302

MOTIVATION:

Fashion is the way we present ourselves which mainly focuses on vision, and has attracted great interest from researchers. Visual classification of commercial products is a branch of the wider fields of object detection ad feature extraction in computer vision, and, in particular, it is an important step in the creative workflow in Product industries. Automatically classifying product features makes both designers and data experts aware of their overall production, which is fundamental in order to organize marketing campaigns, avoid duplicates, categorize apparel products for e-commerce purposes, and so on. There are plenty of products being manufactured and our project focuses on simplifying and making product classification easier thus giving an overall analysis of the production unit.

RESEARCH GAP:

1. **Lack of diverse and large-scale datasets:** Deep learning algorithms require large amounts of data to be trained effectively. However, there is a lack of diverse and large-scale datasets available for clothing image recognition, which limits the accuracy and robustness of the models.
2. **Limited research on real-time clothing recognition:** While deep learning models have shown promising results in clothing recognition, most of the existing research has focused on offline recognition, where images are pre-processed before being analyzed. There is a need for real-time recognition methods that can analyze images in real-time as they are captured.
3. **Interpretability of models:** Deep learning models are often viewed as black boxes, making it challenging to understand how they make their decisions. There is a need for research on how to make these models more interpretable so that they can be better understood and trusted by users.
4. **Limited research on multi-modal recognition:** Clothing recognition can benefit from multi-modal recognition, where information from different sources, such as text descriptions and product specifications, is used to improve the accuracy of the model. However, there is limited research in this area, and more work is needed to develop effective multi-modal recognition models.
5. **Limited research on fine-grained recognition:** Clothing recognition can also benefit from fine-grained recognition, where specific attributes of clothing items, such as the texture or

pattern, are analyzed. There is a need for research on how to develop deep learning models that can effectively recognize these fine-grained attributes.

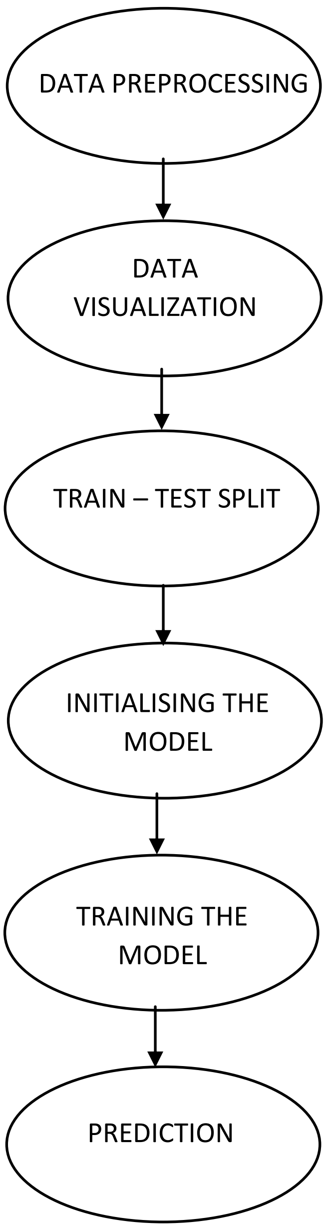
AIM:

The main objective of the project is to use Deep Learning techniques to classify fashion products.

Through the process

- Understanding and implementing various models and comparing accuracies help in picking the right model.
- Understanding Data Visualization techniques to get to know about the dataset.
- Learn the data-cleaning process through which unnecessary data can be removed.
- Learn how to train the neural network and what activation function suits the best.

METHODOLOGY:



DATA PREPROCESSING:

The obtained raw data is filled with missing values and certain attributes that create discrepancies in the final results. Moreover, the dataset has data of values of different sizes. This indirectly affects how much a particular attribute adds imbalanced weight to the result. Hence, we follow certain steps of pre-processing such as normalization, deletion, and standardization to make the dataset suitable for processing in the chosen models.

DATA VISUALIZATION:

Followed by Data Pre-processing, the data is clearly visualized to get a deeper understanding of the chosen dataset. The graphical and pictorial visualization of various attributes is analyzed to further train with the models. This helps us understand the weightage and data split of attributes in the chosen dataset.

SPLITTING THE DATA SET AND GROUPING:

The ImageDataGenerator function is used to receive the original data as input and then randomly transform it, returning an output containing only the newly altered data. The Keras ImageDataGenerator module is also used to augment data in an effort to increase the model's overall generalizability which includes Random translations, rotations, scale changes, and vertical flips. It takes the flow from the data frame and fetches the required attribute from the data frame and then looks for the folder and creates it as a flow of data.

This follows the split of the dataset in the ratio 80:20. The 80% of the dataset is used to train the model and the remaining 20% is used to test the model. The splitting of the dataset is done to avoid discrepancies in the final results due to the redundancy of data in the trained and test dataset.

Upon implementation for the chosen dataset, 35536 images are grouped for training and 8883 images are grouped for testing.

```
Found 35536 validated image filenames belonging to 7 classes.
```

```
Found 8883 validated image filenames belonging to 7 classes.
```

The classes represent the different types of categories present in the chosen attribute.

INITIALIZING THE MODEL:

This step involves building the architecture for the neural network model by using models and layers.

Models are the basic data structure of Keras. A model defines the organization of layers

1.) **Sequential**

This model allows the creation of model layer by layer in a sequential manner. This model is most suited for a simple stack of layers that have 1 input tensor and 1 output tensor.

2.) **Conv2D**

This layer creates a convolution kernel (i.e.) a convolution matrix that can be convolved with an image for blurring, sharpening, etc.

3.) MaxPooling2D

Used to reduce the storage transmission requirement by reducing the spatial dimensions of the output data.

4.) Dense

Contains all the neurons that are deeply connected within themselves. This implements the operation = **activation(input*kernel) +bias).**

These functions were used to create the architecture of the Neural network and build the model.

TRAINING THE MODEL:

This step involves training our model by giving the chosen dataset as input. Using **fit.generator** function, the data for training the neural network is formed. 80% of data is made into epochs for training and 20% of data for validation. The whole dataset is split into 139 epochs and each epoch contains 256 images for training the model. The model is trained 5 times and the accuracy of the model is displayed. Each time the model is iterated, the weights and biases get updated and the accuracy is improved.

TESTING OUR DATA:

This step involves testing by giving random input from the dataset and checking the predicted results by the model. The output is the representation of an array of 7 values where each value represents the probability of the values of the respective classes.

PREDICTION:

The arg max function returns the column name which has the maximum value of probability and predicts the type of input data. The column represents the Mastercategory of the input data given thus predicting the type of input given.

DATASET EXPLANATION:

https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small

The dataset used in our project is “Fashion Product Images (Small). The growing e-commerce industry presents us with this large dataset. It has multiple label attributes describing the fashion product which was manually entered while cataloging. It also has descriptive text comments on the product characteristics.

This dataset contains images of around 44000 products with their labels. Each product is identified by an ID. It contains a map of all the products in styles.csv. It also has some of the key product categories and their display names in styles.csv. Using the masterCategory column from styles.csv, the model has been trained.

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011.0	Casual	Turtle Check Men Navy Blue Shirt
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Summer	2012.0	Casual	Peter England Men Party Blue Jeans
2	59263	Women	Accessories	Watches	Watches	Silver	Winter	2016.0	Casual	Titan Women Silver Watch
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Fall	2011.0	Casual	Manchester United Men Solid Black Track Pants
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Summer	2012.0	Casual	Puma Men Grey T-shirt
5	1855	Men	Apparel	Topwear	Tshirts	Grey	Summer	2011.0	Casual	Inkfruit Mens Chain Reaction T-shirt
6	30805	Men	Apparel	Topwear	Shirts	Green	Summer	2012.0	Ethnic	Fabindia Men Striped Green Shirt
7	26960	Women	Apparel	Topwear	Shirts	Purple	Summer	2012.0	Casual	Jealous 21 Women Purple Shirt
8	29114	Men	Accessories	Socks	Socks	Navy Blue	Summer	2012.0	Casual	Puma Men Pack of 3 Socks
9	30039	Men	Accessories	Watches	Watches	Black	Winter	2016.0	Casual	Skagen Men Black Watch

```
[25] df.shape

     (44424, 11)
```

We are going to predict the masterCategory by implementing the model.

ALGORITHM

Initialize the network architecture: Determine the number of layers, types of layers (convolutional, pooling, etc.), filter sizes, and number of filters for each layer.

Input preparation: Preprocess the input data (e.g., normalize, resize, crop, augment, etc.).

Forward pass: Pass the preprocessed input data through the network to generate a prediction. Each layer performs a specific operation on the input and outputs an activation map.

Loss calculation: Compare the predicted output with the ground truth label to calculate the loss (error) for the current prediction.

Backward pass (Backpropagation): Calculate the gradient of the loss with respect to the weights and biases of each layer. Use this gradient to update the weights and biases of the layers using an optimization algorithm

Repeat steps 3-5 for multiple iterations (epochs) until the network converges to an acceptable level of accuracy.

Evaluation: Test the performance of the trained model on a held-out dataset and calculate metrics such as accuracy, precision, recall, and F1-score.

Deployment: Deploy the trained model for making predictions on new unseen data.

RESULTS:

Epoch = 1

Iteration	Training Accuracy	Validation Accuracy
1	93.06%	93.10%

Epoch = 2

Iteration	Training Accuracy	Validation Accuracy
1	94.67%	93.44%
2	95.22%	94.88%
Overall Accuracy	94.88%	

Epoch = 5

Iteration	Training Accuracy	Validation Accuracy
1	80.27%	90.24%
2	92.71%	93.62%
3	94.62%	94.47%
4	95.29%	95.06%
5	95.84%	95.56%
Overall Accuracy	95.56%	

Epoch = 7

Iteration	Training Accuracy	Validation Accuracy
7	97.84%	95.83%
Overall Accuracy	95.83%	

Epoch = 8

Iteration	Training Accuracy	Validation Accuracy
8	98.20%	97.11%
Overall Accuracy	97.11%	

Epoch = 10

Iteration	Training Accuracy	Validation Accuracy
10	98.66%	97.07%
Overall Accuracy	97.07%	

```
[23] filename = "15970.jpg"
     my_dict = training_generator.class_indices
     key_list = list(my_dict.keys())
     val_list = list(my_dict.values())
     print(key_list[val])

     Apparel
```

id	gender	masterCat	subCatego	articleTyp	baseColou	season	year	usage	productDisplayName
15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011	Casual	Turtle Check Men Navy Blue Shirt

COMPARISON WITH OTHER SIMILAR WORKS:

- In this paper, they have used the Fashion-MNIST dataset. It has 60000 training data and 10,000 testing data. Each data is a grayscale image of size 28x28 pixels. The dataset has 10 output classes. They made three models to predict the data. Multilayer perceptron learning model, convolution neural network model, and extreme learning model.
- With 128 hidden neurons, the MLP is giving 86.5 percent training accuracy and 89 percent testing accuracy. CNN is giving 90.8 percent training accuracy and 90.7 percent testing accuracy. The ELM is giving 96.4 percent training accuracy and 96.3 testing accuracy.

- With 256 hidden neurons, the MLP is giving 88.5 percent training accuracy and 89.9 percent testing accuracy. CNN is giving 91.4 percent training accuracy and 91.5 percent testing accuracy. The ELM is giving 96.5 percent training accuracy and 96.4 testing accuracy.

- With 512 hidden neurons, the MLP is giving 88.9 percent training accuracy and 90.2 percent testing accuracy. CNN is giving 91.7 percent training accuracy and 91.7 percent testing accuracy. The ELM is giving 96.9 percent training accuracy and 96.7

- With 1024 hidden neurons, the MLP is giving 89.0 percent training accuracy and 90.3 percent testing accuracy. CNN is giving 92.2 percent training accuracy and 92.3 percent testing accuracy. The ELM is giving 97.4 percent training accuracy and 97.1 testing accuracy.

- With 2048 hidden neurons, the MLP is giving 88.6 percent training accuracy and 90.3 percent testing accuracy. CNN is giving 92.8 percent training accuracy and 92.7 percent testing accuracy. The ELM is giving 97.5 percent training accuracy and 97.3 testing accuracy.

- With 4096 hidden neurons, the MLP is giving 88.7 percent training accuracy and 90.4 percent testing accuracy. CNN is giving 93.4 percent training accuracy and 93.3 percent testing accuracy. The ELM is giving 97.8 percent training accuracy and 97.5 testing accuracy.
- The design is implemented using an Intel Core i5 processor with a 6 GB RAM capacity and 1 TB hard disk.
- As an operating system, Windows 10 is used. The typical training time for the models with 4096 hidden neurons is 3,685 seconds for MLP, 36,000 seconds for CNN, and 40 seconds for ELM testing

APPLICATIONS OF THE PROJECT:

There are many ways in which the fashion industry can benefit from deep learning-based fashion product classification.

- **E-commerce:** Fashion product classification can be used by online clothing retailers to automatically sort items into relevant categories. As a result, the shopping process is simplified and expedited for the customer.

- **Visual search** is made possible through fashion product categorization on e-commerce sites. Customers can now upload images of products they like and have website display options that are visually similar to those images.
- **Inventory Management:** Deep learning algorithms can be used by the fashion industry for inventory management. This aids in lowering the potential for either excessive or insufficient stock levels.
- **Trend Analysis:** The newest style trends can be analyzed with the help of deep learning algorithms. This can be useful when deciding what products to stock because it allows you to anticipate which ones will be popular in the near future.
- **Customization:** Customers can be given specific suggestions based on their preferences by using fashion product classification. This means that shoppers are only shown products that are relevant to them, boosting conversion rates.

CONCLUSION:

The dataset is pre-processed and then clearly visualized. Then, the neural model is built and the input image is predicted correctly. The model is iterated 8 times (epoch = 8) to attain maximum accuracy of 97.11%. The model predicted the output of Mastercategory of the chosen input when validated on the built neural model.

REFERENCES:

1. Samia, B., Soraya, Z., & Malika, M. (2022, May). Fashion Images Classification using Machine Learning, Deep Learning and Transfer Learning Models. In 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA) (pp. 1-5). IEEE.
2. Bhoir, S., & Patil, S. (2022, April). Transfer Learning with Deep Neural Networks for Image Classification in the Ecommerce Industry. In 2022 IEEE 7th International conference for Convergence in Technology (I2CT) (pp. 1-8). IEEE
3. S. Shubathra, P. Kalaivaani and S. Santhoshkumar, "Clothing Image Recognition Based on Multiple Features Using Deep Neural Networks," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 166-172, doi: 10.1109/ICESC48915.2020.9155959.
4. Nicherala, Y. K., Sadula, S., & Shrinivas, V. P. (2022, August). Deep Learning Based Sustainable Material Attribution for Apparels. In 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE) (pp. 1352-1357). IEEE
5. Seo, Y., & Shin, K. S. (2019). Hierarchical convolutional neural networks for fashion image classification. Expert systems with applications, 116, 328-339
6. Kolisnik, B., Hogan, I., & Zulkernine, F. (2021). Condition-CNN: A hierarchical multi-label fashion image classification model. Expert Systems with Applications, 182, 11519
7. Vijayaraj, A., Raj, V., Jebakumar, R., Gururama Senthilvel, P., Kumar, N., Suresh Kumar, R., & Dhanagopal, R. (2022). Deep Learning Image Classification for Fashion Design. Wireless Communications and Mobile Computing, 2022

Individual role::

My role about this project is data processing. The data's from this project are collected from Kaggle site as it has more accurate data on them. We collected the fashion product data from ~~the~~ that site and performed data pre-processing. After this, the data is cleaned by removing unnecessary values & comma seperated values. Then in that dataset each dataset has some unique ID values & their ~~corresponding~~ corresponding classification of products. We have converted those ID values to images inoreder to classify the image of the corresponding product easily. Those image values are saved in corresponding ID values .JPG format based on the classification. ~~For example~~ Then we have done CNN algorithm to identify the product that we given as input. Here we have done upto 8 epoch where in each epoch accuracy of the model increases. After 8 ~~epoch~~ epoch the accuracy of the model ~~dec~~ decreases due to the overfitting of data. Here we have used CNN algorithm to get the data without any losses by reducing the dimension for the given image. Here we have JPG image for each ID classification in order to use the ~~K-Means~~ CNN ~~clustering~~ effectively. By doing this the accuracy of our model increases. Here we have done convolution to get good clarity of the images. Here we have increased the accuracy by increasing the no. of hidden layers in the model. Here the MAX Pooling 2D is done to reduce the storage transmission requirement by reducing spatial dimension. We have taken 80% of data as training & 20% of data as testing. The size of the dataset is (44424, 11). 44424 - rows 11 - columns. The whole dataset is splitted into 139 epochs & each epoch contains 256 images for training the model. Each time model iterated, weights & bias get updated & the accuracy of the model gets improved. For testing data, we have taken random i/p. The O/P is the representation of an array of 7 values where each value represents the probability of the value of the respective class. The arg max function returns the column name which has the maximum value of probability & predicts the type of i/p data.