PART A: FOR APS360 PROJECT

Danial Khan
Student# 1007088081
dani.khan@mail.utoronto.ca

Suleiman Najim
Student# 1006662504
suleiman.najim@mail.utoronto.ca

Payas Hasteer
Student# 1006823751
payas.hasteer@mail.utoronto.ca

Abhi Sharma
Student# 1006700438
abhib.sharma@mail.utoronto.ca

1 INTRODUCTION

Ecommerce is one of the biggest industries in the world today, and online apparel shopping has been at the forefront of this exponential rise of online eCommerce. Although eCommerce has made people's lives easier and more convenient, the process can still be enhanced, particularly when searching for products.

Often, people scroll through social media and see their friends or a celebrity wearing the perfect outfit. The problem is that the brand or product name can be unknown; therefore, finding it online can be daunting. To combat this, we created a machine learning model that takes an image of apparel as the input and predicts the category of the clothing/apparel which can be used to recommend the same or similar products. We believe machine learning was the best and only way to accomplish this goal, as this is an image classification problem. We cannot explicitly program the model to predict a clothing piece in an image as there are too many edge cases, so the best way is to use ML and let our model learn from a dataset so that it can predict images that have never been seen before.

2 ILLUSTRATION

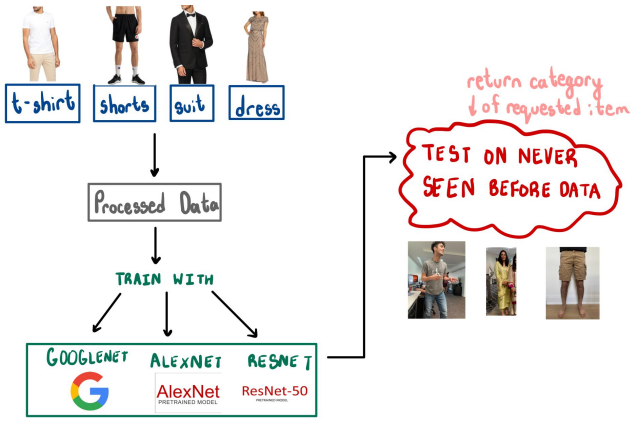

Figure 1: Illustration of the model.

3 BACKGROUND AND RELATED WORK

Fashion category and attribute prediction has been a popular topic in machine learning as it has many use cases, mainly regarding searching, as mentioned earlier in the introduction. We have researched related work in this field to get a better understanding of the problem and thus reach the best possible solution for our model.

Before CNN, earlier approaches used feature extractors like HOG and SIFT to classify clothing attributes. Next, some models utilized CNN, like FashionNet, which uses a CNN architecture similar to VGG-16 to predict attributes and categories. Additionally, R-CNN models have been utilized for body detection to generate clothing proposals (1).

Moreover, there have been some advancements with cross-domain clothing detection, where we learn a transformation that aligns the source and target representations into a common feature space. Earlier models have also used deep networks to extract apparel attributes from the input images to create a visual embedding of clothing style to help study fashion trends worldwide (1).

4 Data Processing

For our project, the quality of data was vital in differentiating the variety of categories we have created. While searching through various possible department apparel websites, we learned that most websites have tight restrictions for automated data collection. We searched several websites and found the Myers Department store had no firewall preventing data collection(2).

To extract the data from the department store, we decided to use web scraping techniques,

1. An array of category strings were compiled to differentiate the types of images present in the model.
2. Each category was attached to a generic string for the Myers department store search function.
3. To extract all the images from the search results, Beautiful Soup was used by finding all 'img' HTML elements with the items 'src'
4. All images were placed in a two D array separated by each category and placed into google drive into training, validation, and testing folders in a 60 20 20 split, respectively.

Figure 2: Sample images for each category with their labels.

As seen in the backgrounds, most pictures either have a white or light gray background to contrast the item. In total, 11643 images were collected from the Myers Department Store website. The breakdown on how the images were separated is found below:

Table 1: Large dataset image breakdown

Type of Data	Number of Data Samples
Training	6988
Validation	2327
Testing	2328
Total	11643

5 Architecture

5.1 AlexNet

The neural network finally chosen was Convolutional Neural Network (CNN) with transfer learning using Alexnet. Using Alexnet with transfer learning enables fast pre-training of our extensive image

classification dataset of 11.6k images. This enabled the model to learn the generalization of different clothing types across the classes. Alexnet itself consists of 8 layers, with 5 being CNN and three being fully connected, taking in the input of images of 256x256 in 3 channels of RGB. Alexnet also utilizes a dropout layer to mitigate overfitting during training and carryover into the forward pass (3).

The CNN transfer model (fig. 2) begins with one convolutional layer taking in 256 x 256-pixel images, with channel size 256 with kernels of size 3 and a padding of 1, followed by a 2-dimensional max pool of kernel size 2 and stride 2. This pooling layer is then followed by two fully connected linear layers, with the first taking an input of size 256x3x3 and output of size 32, and the second taking in 32 as input and outputting a feature of size 9. The prediction for the feature is then based on a ReLu activation function. After this, Cross entropy and a Stochastic Gradient Descent (SGD) is used in training to calculate the loss and optimization, respectively.

The CNN with transfer learning was chosen as the optimal neural network based on experimentation against a standalone Recurrent Neural Network, 2D, and 3D CNN model.

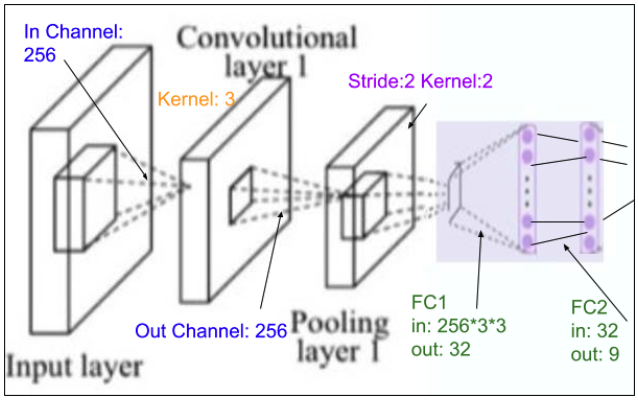

Figure 3: Transfer CNN Model Architecture(3).

Table 2: Hyperparameter Tuning for Transfer CNN Model	
Hyperparameter	Value
Batch Size	27
Learning Rate	0.001
Epoch	20
Momentum	0.9

5.2 GoogleNet

The Googlenet Architecture consists of the same Transfer CNN used for Alexnet, with the same Hyperparameter tuning shown in table . GoogleNet comprises 22 layers of a combination of 1x1,3x3 and 5x5 convolutions and 27 pooling layers for those combinations. GoogleNet is used once again to pretrain the images used in the Transfer CNN model with the same hyperparameters in table 2.

5.3 ResNet

The Residual Neural Network comprises 50 CNN layers and utilizes skip connections to mitigate a vanishing gradient in the multiple-layered CNNs. This allows the gradient to skip through the network layers optimally without negatively impacting the model's performance. ResNet has several 1x1 convolutional layers, which prevents more additional parameters carried forward. ResNet is used along with the same transfer CNN and parameters in table 2.

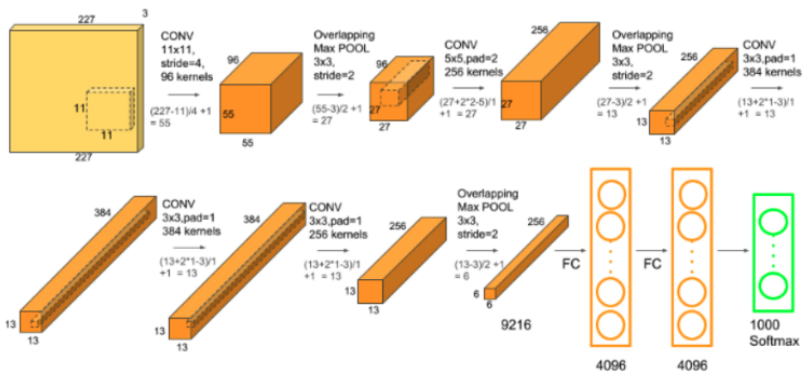

Figure 4: Image of Alexnet Architecture used in pre-training of dataset(3).

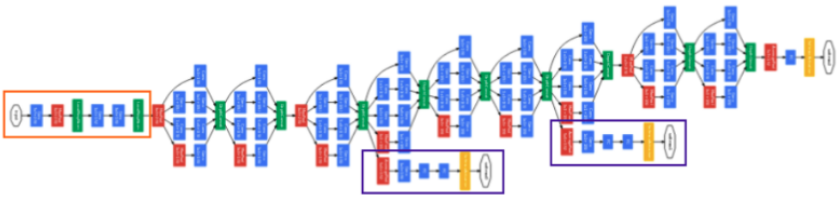

Figure 5: Illustration of GoogleNet Architecture.

6 Baseline Model

The baseline model used is a CNN with 2 CNN layers taking input channel of size 3, output channel of size 5, and kernel size 5, with one max pool layer of size two and stride 2, followed by the second CNN layer with input channel size 5, output channel size ten and kernel size 5, finally follows by two fully linearly connected layers. Two ReLu activation functions are used in the forward pass additionally. The baseline model was trained, validated and tested against a handmade data split as shown in the table below, with a split of roughly 44%, 21% and 35%, respectively. The hyperparameters used for the baseline followed the format shown in table

.

Table 3: Small dataset image breakdown	
Type of Data	Number of Data Samples
Training	147
Validation	70
Testing	118
Total	335

Table 4: Hyperparameter Tuning for Baseline Model	
Hyperparameter	Value
Batch Size	27
Learning Rate	0.001
Epoch	110
Momentum	0.9

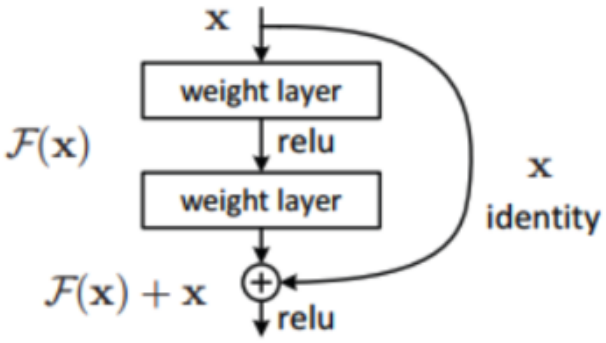

Figure 6: Illustration of ResNet Architecture.

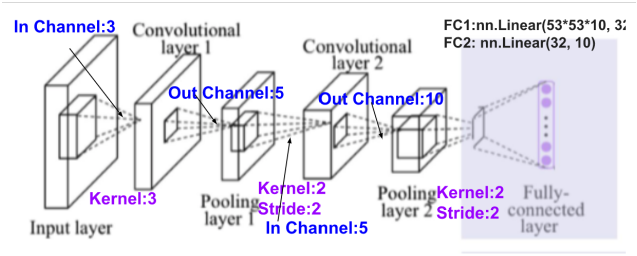

Figure 7: Model Architecture for Baseline

7 Quantitative Results

Since our project classifies different images into clothing categories with similar attributes, this is an example of a classification problem. Therefore, we decided to rely on Cross entropy as our loss function for training to measure the performance of our model. The reason we went with Cross-Entropy instead of MSE, is because, unlike cross-entropy, MSE loss functions do not punish incorrect classification severely which makes it a good tool for regression but not classification problems that deal with discrete data and not continuous(4).

We used the loss/iteration graphs, training, validation, and testing accuracy to measure the performance of our models quantitatively. The Baseline was over-fitted with a small dataset and the other models use the full dataset.

8 Qualitative Results

The following presents all the categories along with a sample image with our best model outputted using the AlexNet implementation and achieving a test result accuracy of 98.3050%. To demo our model, we passed in one photo per category and compared the prediction with the label and the confidence level of that prediction. The following presents our results:

The category that our model was least confident in is the men's t-shirt, with a confidence level of 91.02%. Since women's dresses can have similar content, it makes sense that the confidence level decreases. Additionally, some of the t-shirt data also included images of men wearing shorts, which sometimes creates a disparity in the results.

The model performed exceptionally well on watches, sunglasses, and men's shorts. The image features of these three categories are pretty distinct, with men's shorts, legs, and watches having remarkably similar characteristics compared to other watches. We believe that the data quality was

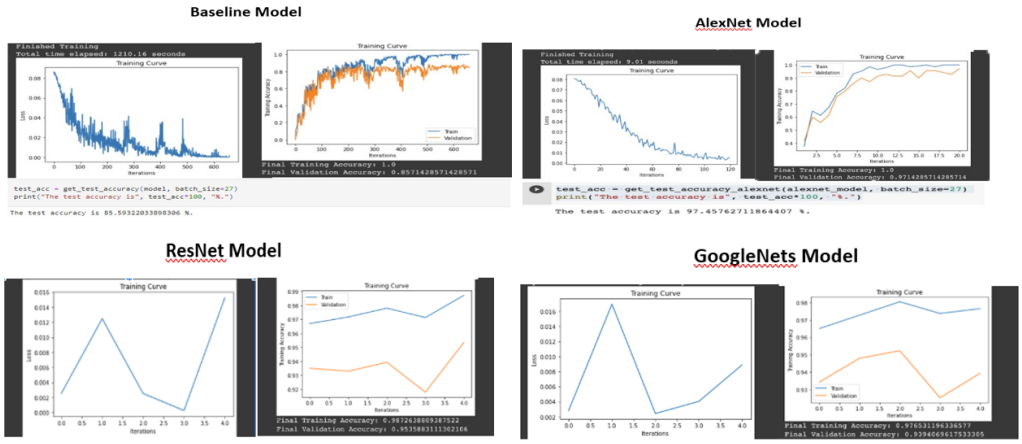

Figure 8: Baseline, AlexNets, Res50 and GoogleNets Results

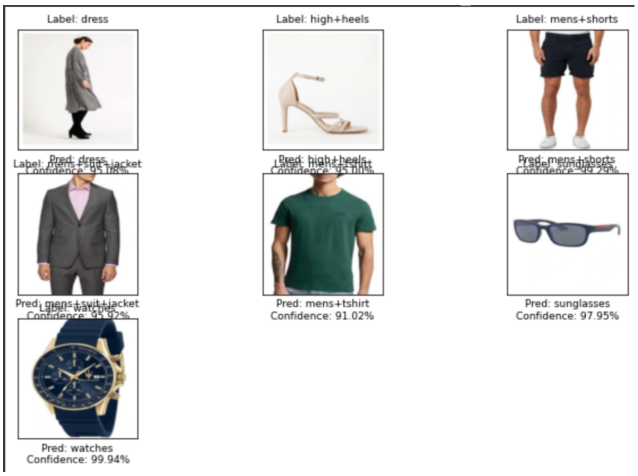

Figure 9: Prediction of category for different clothing/apparel

critical in helping train our model because instead of discerning the image's background, the model only needs to focus on the features of the object we want to classify.

9 Evaluate Model on New Data

New Model Data Collection and Processing:

1. Data was collected as a combination of first-hand pictures of our own clothing, and some images from google search.
2. The images were taken in different environments, with an equal frequency of images amongst the group members' clothes so as to mitigate any advantage or disadvantages given by an image's particular background when testing the robustness of the model in classifying clothes.
3. The images were then cropped and resized to be 224x224 RGB images and had their respective labels associated with them for testing on real new data.

To give a strenuous test to our model, the team used different backgrounds for each image to see how the model could handle other circumstances. The pictures with the wrong predictions are the dress

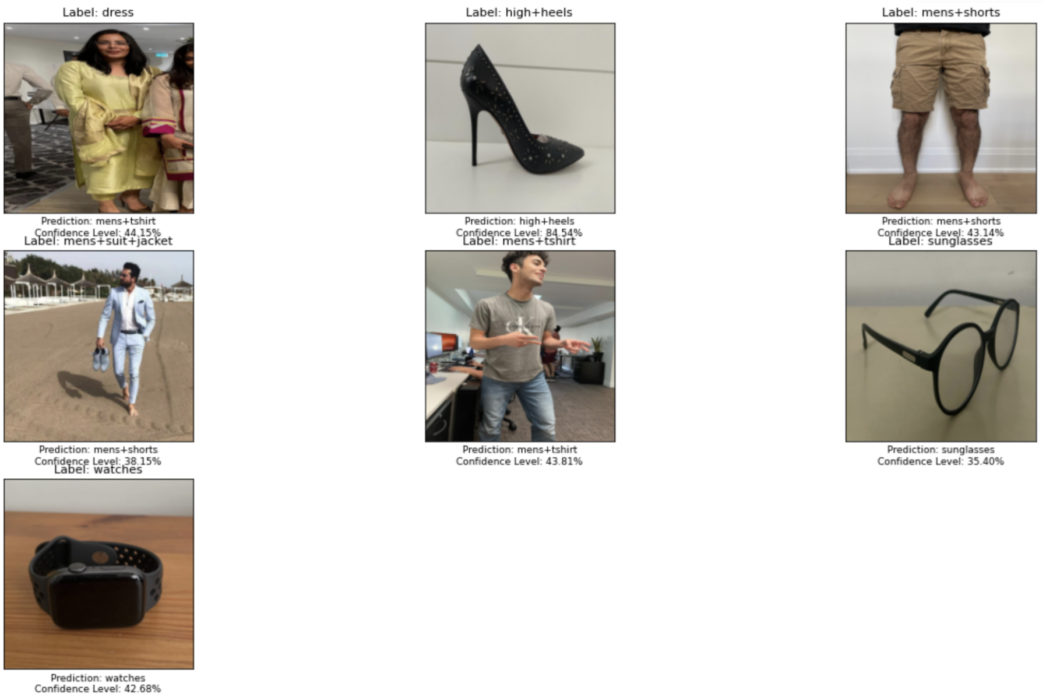

Figure 10: Image of results for our new data

and the men's suit examples. However, our model could predict most images correctly with a lower confidence level than our testing data. It is due to the nature of the sample images and factors like background, orientation and quality, which we will discuss in more detail in the discussion section.

10 Discussion of Model

Table 5: Comparison between Online Data and New Handmade Data

Class	Online Data Confidence	Handmade Data Confidence
Dress	95.1%	44.2%(Incorrect Label)
High Heels	95.0%	84.5%
Shorts	99.3%	43.1%
Suit	96%	35.4%(Incorrect Label)
T-Shirt	91%	43.8%
Sunglasses	97.9%	38.2%
Watches	99.99%	42.7%

The new data set analysis using our best model AlexNet with the Transfer CNN, showed that the testing accuracy resulted in 98.3% from online data. However, with handmade data, the model had lower confidence in all the classes (fig 9) relative to the online training data. These results were expected since the quality of images taken by the group members was significantly lower than the typical production quality of clothing images found online in terms of image resolution, background colour, lighting, clothing positioning, and general editing. We determined that positioning was a leading factor in the discrepancy between online and handmade images, which disfigured the typical clothing dimensions of the model. Despite this, the model did perform relatively well in

classifying handmade items such as ”high+heels,” ”sunglasses,” and ”watches” in the handmade class with confidence levels of 84.54%, 38.2%, 42.7% compared to the online testing accuracies of 95%, 97.9%, 99.9% respectively. The reasoning behind why these percentages fared better amongst the handmade class was because the physical structure of these items is distinct from other clothing items. This is supported by how ”dress” and ”mens+suit” were incorrectly labelled as ”mens+tshirt.” In hindsight, having trained the model on images of clothing items with varying 45-degree left and right angles in addition to the front view with a human model would have increased the testing accuracy of the model on real-world data since production quality images do not accurately reflect images of clothing items in real-world situations. However, the caveat with this goal would be finding large enough datasets with real-life hand-taken images in such angles to train on, although the team could have generated a 1000 image dataset with such angles in our clothes for training purposes on the AlexNet Model. With all things considered, we believe the model performed well for the unseen real-life data.

Based on results from the new data set, image classification of clothing items that are stretched/compressed/wrinkled or distorted by the human and environment, e.t.c adds a higher level of complexity, making this the biggest reason for poor classification. Past research on the same problem of clothing identification(1) has used datasets of runway models to simulate the realistic application of the ML model on human beings in motion while also using a Recurrent CNN model with a manually decaying learning rate and smaller batch size in their model to optimize training. Additionally, image resolution adjustments could be made on all data such that more minor nuances on clothing due to fashion trends (for example, a zipper on a t-shirt) could be applied so that the model gets the bigger picture.

In this project, we learned how to successfully web scrape data efficiently, build and train an efficient CNN model with transfer learning and various architectures, tune hyperparameters efficiently and test the model on real-world data to drive accurate results.

11 Ethical Considerations

The training data utilized to train the project’s model doesn’t trigger any potential privacy issues, as the data was widely available on online platforms and was used at the time of web scraping. As with many other ML predictors, a qualitative quotient for efficiency is directly linked to adaptability and versatility. Keeping in mind, the unpredictable nature of the real-world causes such trends to often change. Focusing on this, the model now avoids all gender bias by including apparel that tends to all genders.

Moreover, mainly basing the training data on Myers, the model can directly be affected by the trends and biases that can come with it. It can tend to deviate from the tendencies mentioned above, create unwanted prejudice, and make the model flawed. Acquiring training data from numerous online platforms is the solution.

Additionally, there’s another ethical issue: adaptability bias. The model would not be able to tackle the problem of showing a holistic apparel review but would be significantly affected by the current trends as our training data is based on platforms that cater to fashion trends. Hence, it does not show obsolete trends, aka results.

12 Project Difficulty

The main problem the team faced while writing the model was Web Scraping. The team was blocked by 12 website firewalls while automating. Finally, the team managed to find Meyers to automate web scraping for training, validation, and testing. Moreover, figuring out the best model to use was a strenuous task and required immense brainstorming and extensive coding.

REFERENCES

[1] M. Jia, Y. Zhou, M. Shi, and B. Hariharan, "A deep-learning-based fashion attributes detection model," 2018.

[2] "Myers department store," Aug 2022.

[3] V. Kurama, "A guide to alexnet, vgg16, and googlenet," Apr 2021.

[4] "What is the different between mse error and cross-entropy error in nn," Sep 2017.

https://www.overleaf.com/project/62f8a329734b7d8a2ec41a48