Price Prediction Tool

(CNN & Regression Model)

Bootcamp Group#4:

Christina Leung
Ismail Omer
Jacky Zhang
Yug Sharma

Purpose

- Our tool is designed for general consumers

- Imagine yourself walking, see someone with good fashion
  - What is that clothing and how much does it cost?
  - Take a picture into and input into our code

Background

Libraries used:

- Pandas/Matplotlib
- TensorFlow
- PyTorch/TorchVision

TensorFlow is an end–to-end open source platform for machine learning

PyTorch is a machine learning framework used for image recognition

Background

Database: We used the MNIST(Modified National Institute of Stands and Technology) database

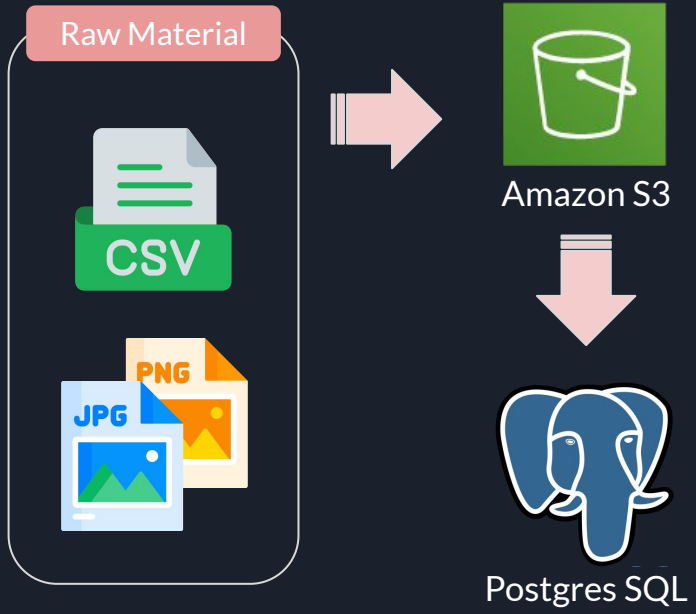
- Large dataset but small image resolution

We use PostgreSQL (local) for ETL

Methodology

- Analyzed original clothing dataset to see if prices can be predicted
- Colour prediction
- Category prediction
- Brand prediction

ETL Introduction



All of the raw materials are loaded into the AWS S3 bucket for cloud storage.

Through setting up AWS Command Line Interface (CLI), raw data would be extracted.

Extracted data would be then loaded to our local Postgres SQL database, and ready for the model.

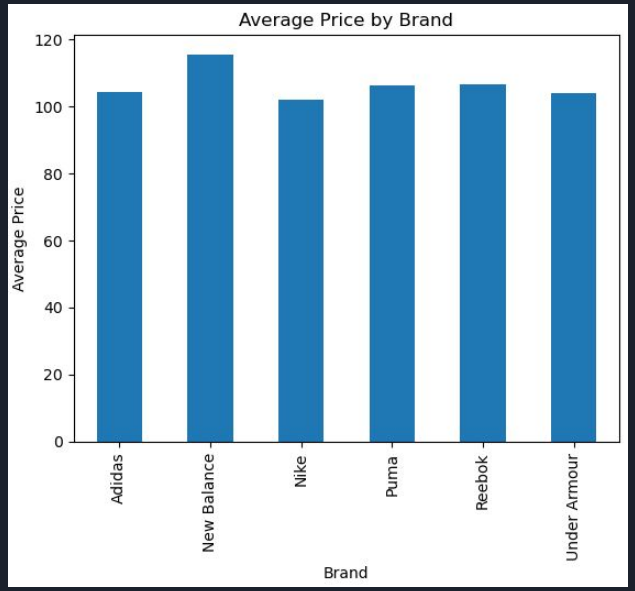
**** Cloud database would be more efficient for the project for further development.*

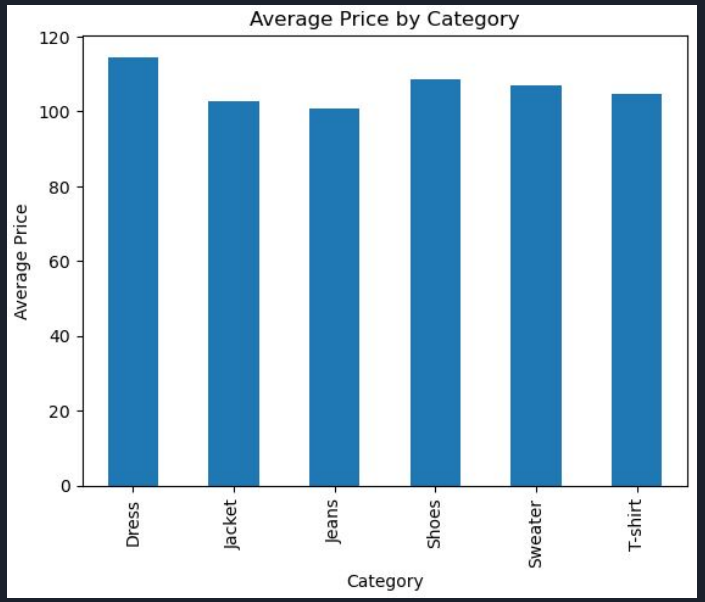
Introduce Dataset

- Clothes Price Prediction CSV from Kaggle
- It includes various features related to clothing items along with corresponding prices
- Features include Brands, Category, Size, Material, Price

	Brand	Category	Color	Size	Material	Price
0	New Balance	Dress	White	XS	Nylon	182
1	New Balance	Jeans	Black	XS	Silk	57
2	Under Armour	Dress	Red	M	Wool	127
3	Nike	Shoes	Green	M	Cotton	77
4	Adidas	Sweater	White	M	Nylon	113

Exploratory Data Analysis





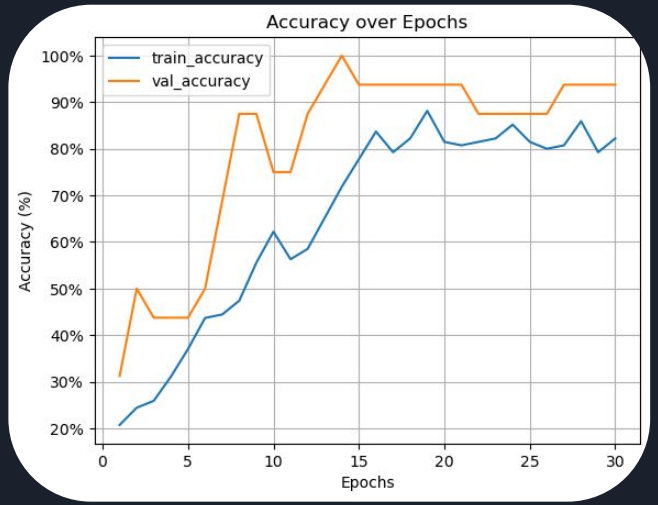
Convolutional Neural Network

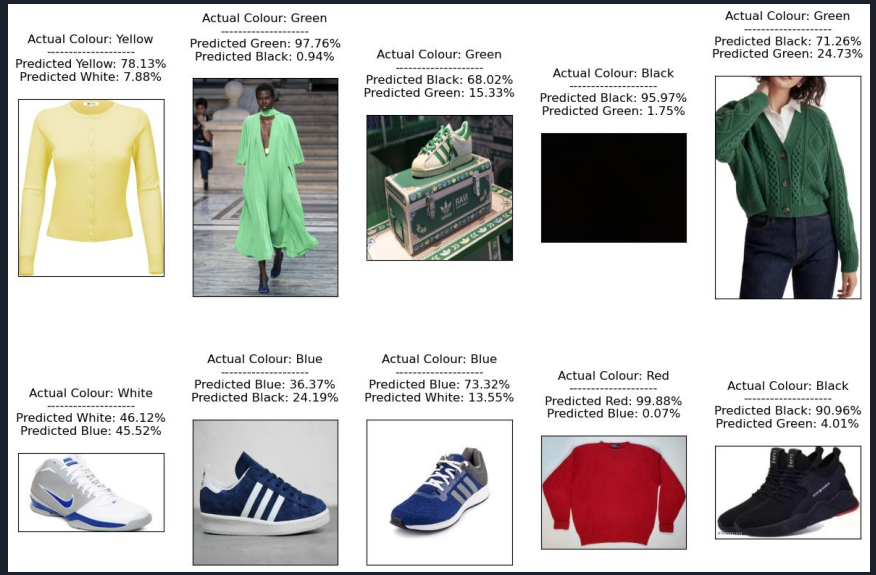
Colour and Categories

- PyTorch
- Colours used in our fashion dataset (black, blue, green, red, white, yellow)
- Category (T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, shoes, bag, and ankle boot)

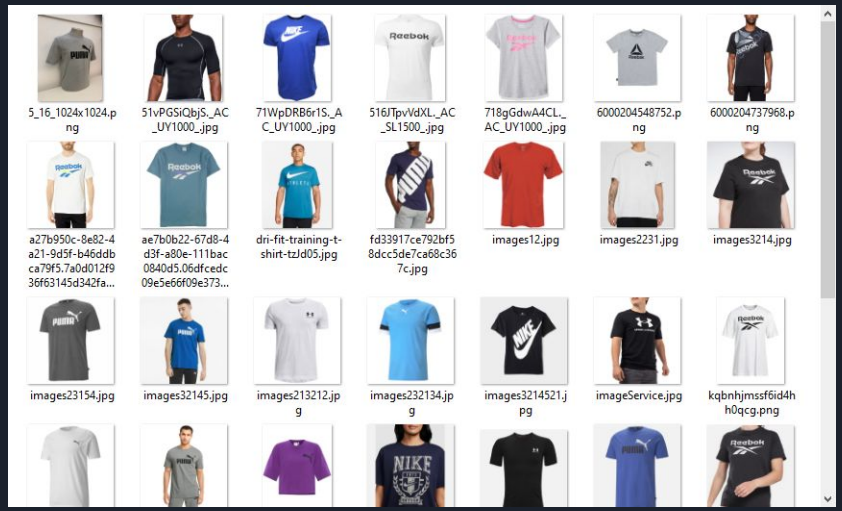
Colour CNN Analysis

- Simpler/faster model of the two
- Smaller dataset 168 images
- 3 neurons





Category CNN



Originally:

- Download own images for categories
- Too many variations for small dataset (250)
  - categories overlap (dress looks like shirt)
- Could only have 50% accuracy at most changing hyperparameters

Category CNN

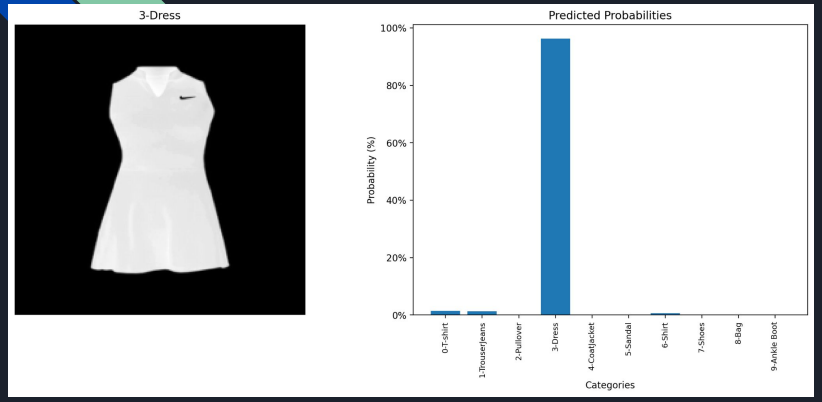
Model:

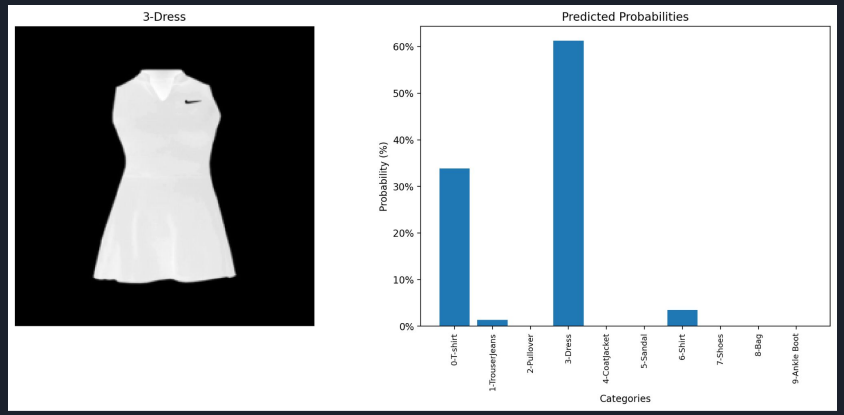
- MNIST Dataset
- 70,000+ black and white images of 28x28 images
- 3 neurons
- 10 categories instead of 6 (no problem as dataset categories subset of MNIST)

Category CNN

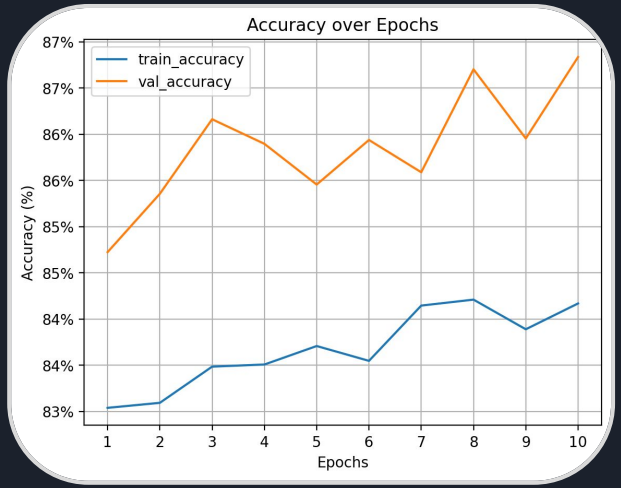
- User image converted to black and white, then inverted
- Image resizing down to 224x224
- MNIST images upsized from 28x28 to 224x224

Category CNN Analysis





- 23 minute runtime (2-3 minute/ epoch)
- 83-84% accuracy
- Probabilities change depending on hyperparameters (epochs, neurons)
  - Small epoch = probabilities more spread



CNN Model for Brand Prediction

(Small scale of training dataset version)

```
# Test the model with a new image
image_path = 'data/test/22.png'
predict_brand(image_path)
```

```
[23]  ✓ 0.4s

...   1/1 [==============================] - 0s 129ms/step
      Predicted Brand: Nike
      Confidence: 0.81875855
```

- Only 20 training images are used for each brand.
- Fast testing speed and high accuracy for internal resources.

Limitations

- Premade MNIST training data
    - i.e.) a pair of shoes in an image or images not in the right configuration
- Not optimized: run-time 2-3 min/epoch `Model Runtime: 23 minutes 28.93seconds`
    - Upsizing 28x28 to 224x224, increasing runtime 100x
- Low # epochs = unable to train all data properly
- Fixed categories MNIST training dataset

Limitations

- Attempted to create a model to predict prices
  - No correlation between the price and other features
  - additional dataset may provide correlation
- Brand training image prediction manually cropped from Internet
  - time consuming to scale it up.
- AWS Free Tier limitation.
  - S3 Bucket Free Tier <= 2000 Request;
  - AWS Database do not have a free version.

Conclusion

- More dataset required to find correlation between features and price

- Colour CNN model works well with over 80% accuracy

- Category CNN model works well only (over 80% accuracy) with B/W images and shapes closely resembling MNIST dataset