P08: Quality assessment of fashion returns in e-commerce using machine learning techniques

Project Team Members: Berk Ali Cam, Muhammad Raza, Alua Kadyrbek - Project Sponsor: Jonas Zeuner

Aims of the project

- Collecting a set of fashion return images
- Pre-processing of the collected data and labeling the dataset according to their quality
- Building a machine learning model
- Evaluate the model's performance using different metrics
- Integration to the web

Examples of the collected data

Clean | Stained | Torn | With pet hair

- The **Data augmentation** library like "**imgaug**" was used to enhances the model's performance by expanding limited training data through image transformations like cropping, flipping, and adjusting contrast and brightness.

CNN Layers

1. **Convolutional layers** extract image features (height, width, and depth) in 3D
2. **Pooling** achieved through max-pooling and mean-pooling, reduces feature sizes
3. **"Global Average Pooling"** was used instead of flatten for parameter reduction.
4. **Fully-connected layers** make predictions.
5. To prevent overfitting, use **L1, L2 regularization**
6. **Dropout** for robust learning.

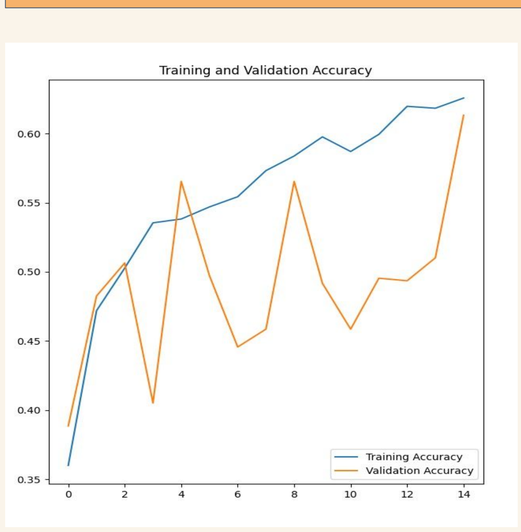
Optimization & Loss Function

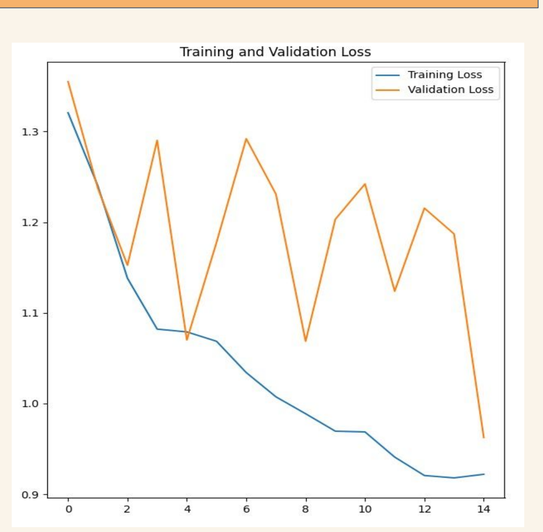
1. **The Adam optimizer** is a robust, gradient-based optimization method suited to nonconvex optimization and machine learning problems.
2. **The loss function** measures the difference between predicted and actual values, allowing the network to learn optimal feature representations. Most popular loss function in use is "Sparse Categorical Cross Entropy", for categorical classification.

Results

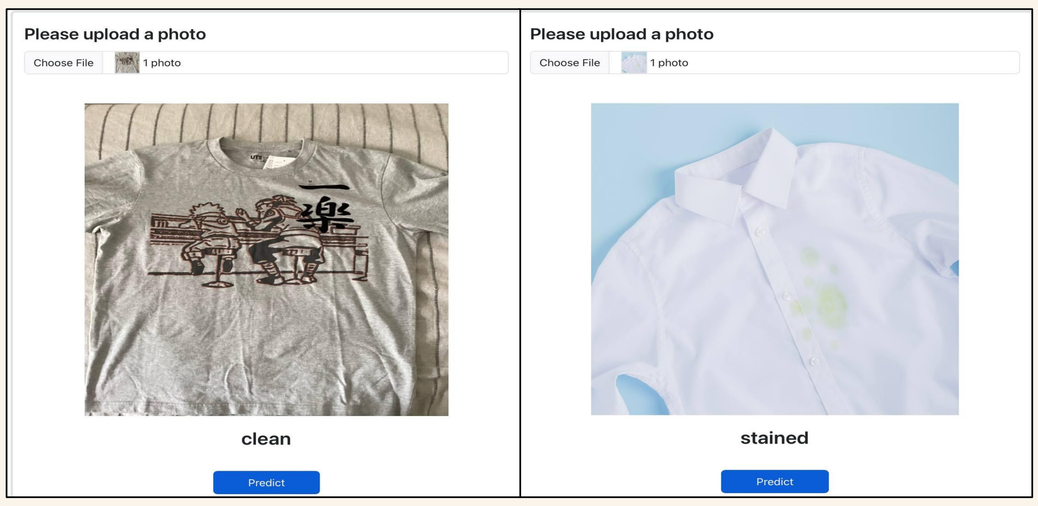
Layer (Type)	Output Shape	Number of Parameters
Conv2D	(None, 512, 512, 3)	1792
MaxPooling2D	(None, 512, 512, 3)	0
Dropout	(None, 512, 512, 3)	0
Conv2D	(None, 512, 512, 3)	73856
MaxPooling2D	(None, 512, 512, 3)	0
Dropout	(None, 512, 512, 3)	0
Conv2D	(None, 512, 512, 3)	295168
GlobalAveragePooling2D	(None, 512, 512, 3)	0
Dense	(None, 512, 512, 3)	263168
Dropout	(None, 512, 512, 3)	0
Dense	(None, 512, 512, 3)	4100
Total Parameters: 638 084		

Validation Results





Web Application



Conclusion

After looking at the validation results, it can bee seen that loss value remains around 0.9, meanwhile accuracy is at around 65%, after 15 epochs. To make the model more robuts, and achieve reliable results, dataset size can be increased.

On the web integration side, model can be used, after converting the model to web-compatible version using Tensorflow.js library, without any problems.