

**RMIT University**

**COSC2753**

**Machine Learning**

**Group Machine Learning Project**

**Tutorial: 3**

**Group:** 4

**Student #1 name:** *Tran Truong Son* (S3818468)

**Student #2 name:** *Vu Thien Nhan* (S3810151)

**Student #3 name:** *Tran Anh Bao* (S3821266)

**Student #4 name:** *Le Anh Dung* (S3915085)

**Student #5 name:** *Nguyen Phat Dat (S3894433)*

**Academic supervisor: M**r. Bao Dang

**Submission Date: 19th May 2024**

**Table of Content**

[1. Abstract: 2](#_Toc1642284561)

[2. Introduction: 3](#_Toc1826705160)

[3. Methodology: 3](#_Toc1600401219)

[a. Data preprocessing: 3](#_Toc41095282)

[b. Literature review for algorithm selection: 3](#_Toc49667699)

[4. Results and Evaluation: 4](#_Toc1625505356)

[5. Conclusion: 6](#_Toc345229547)

[6. References: 6](#_Toc121089738)

[7. Appendices: 6](#_Toc254538664)

# Abstract

The whole process of making a machine learning system that uses advanced analytical methods to solve real-world problems is described in this paper. The goal of this project was to build, test, and improve a machine learning model that could recognize and sort pictures of furniture into specific groups, which would make shopping on an e-commerce site easier. To do the study, different types of furniture images had to be preprocessed, the right machine learning methods had to be chosen and explained, and hyperparameters had to be tuned to make the models work better. The main results showed how well the models we chose were at correctly classifying pictures of furniture and suggesting similar things based on pictures uploaded by users. Based on a mix of accuracy measures and real-world applicability, our final decision was to suggest the best-performing model. Our project's conclusion confirms that our machine learning solutions can improve the user experience in online shopping settings. This gives us a solid base for future improvements and real-world use.

# Introduction

Digital technologies have changed the way stores work, especially when it comes to online shopping. Some industries have grown a lot because of online shopping. One example is furniture stores. Helping customers find exactly what they want quickly and easily is still a big problem when it comes to making shopping better. This project uses machine learning to sort and suggest furniture based on how similar they look, which makes it easier to find what you're looking for in huge online furniture stores.

We can't say enough about how important this problem is in the real world. Customer engagement and happiness are very important in today's fast-paced digital market. A system that can correctly read pictures uploaded by users and suggest related furniture items improves the user experience, which could lead to more sales and customer loyalty. A system like this also makes it easier on users' brains by reducing the need for them to do text-based searches and scroll through many pages of goods that don't apply to them.

The project's goals are threefold: first, to create a machine learning model that can put pictures of furniture into predefined groups; second, to make a recommendation system that uses user-uploaded pictures to find and suggest similar furniture items; and third, to make the model smarter by adding the style of the furniture, which would make the suggestions even better for people who want to make a bigger difference. With these goals, the usefulness and ease of use of e-commerce platforms will be improved, making them more intuitive and in tune with the needs of modern customers.

# 3.Methodology

## a. Data preprocessing:

Since the *Furniture\_Data* folder already came with classified folders for furniture categories and styles, we will be using that as the way to define the furniture's categories and styles. To do that, we first create two arrays for categories and styles, then we use the *os.path.join* method from python to set the directories for each category and style. The images in each of the style folders are then added to an *img\_array* and have their pixel value divided by 255, this is done to normalize the pixel’s RGB (red, green, blue) value to a float with the range of 0 to 1 to ensure consistency.

## b. Literature review for algorithm selection:

In the paper done by P.C. Sen et al. [1], they reviewed several algorithms for both binary classification and multi-label classification, specifically, Decision Trees, Support Vector Machines (SVM), Naive Bayes, and k-Nearest Neighbors (kNN). In the paper done by L. Chen et al. [2], they reviewed the Convolutional Neural Networks (CNNs).

* Decision Trees: Its strength is in its simplistic nature, its ability to quickly compute, as well as its ability to handle categorical data, requiring less data preprocessing. However, the algorithm tends to lead to an unstable model.
* SVM: An algorithm favored for handling image classification task, can handle categorical data, has a consistent high performance. However, it requires a substantially large dataset, which causes long computing time, as well as performance degradation when the dataset contains noisy data.
* Naive Bayes: An algorithm stated as “simple but surprisingly powerful algorithm for predictive modeling”, requiring only a small dataset to be effective which makes it an intensely fast classifier algorithm. However, with its simplicity and fast computing speed, it is known to be a bad estimator compared to other algorithms.
* kNN: Has the advantage of being able to handle large and noisy dataset, as well as being simple to implement. However, this algorithm requires a high computation time and cost.
* CNNs: This is the most adept model for image classification tasks, able to achieve a high accuracy score. However, this algorithm also requires a high computation time and cost.

After doing research and literature reviews, the final decision was to choose between using SVM or CNN to be implemented for our image classification and recommendation model due to both of their reputation as being competent image classifiers. In the end, our team chose to use CNN due to it being able to learn more complex features from images [3].

# 4. Results and Evaluation:

**Evaluation Framework**

The dataset was divided into training and testing sets with an 80-20 split to ensure the model's generalization capabilities were properly evaluated. The primary metric for Task 1 was accuracy, which measures the proportion of correctly classified images. For Task 2, cosine similarity scores were used to evaluate the match between the input image and recommended items, while Task 3 additionally assessed the style consistency of the recommendations.

To validate the robustness of our models, we employed k-fold cross-validation, where the training data was split into k subsets, and the model was trained k times with different validation sets. This approach provided a more reliable estimate of model performance. An independent evaluation was conducted by comparing our results with similar works in the literature and using a separate validation set collected outside the original scope of training and testing.

**Task 1:**

After training the model 3 folds, the results returned are:

* Fold 1: loss of ~1.235, accuracy of ~0.811.
* Fold 2: loss of ~0.932, accuracy of ~0.831.
* Fold 3: loss of ~0.978, accuracy of ~0.817.
* Average loss of ~1.048, average accuracy of ~81.995%

The results showed a consistent level of accuracy at around 82%, which is an acceptable level for a machine learning model, but with the loss value still a great distance away from 0, the model can still be improved further.

**Task 2:**  
After training the model through 15 epoch cycles, the accuracy started rising from 0.355 at Epoch 1, up to 0.978 at Epoch 15, with the most noticeable increases from cycle 1 to 8. The subsequent cycles can be seen peaking at 0.978 accuracy value in Epoch 15, but with slower growth rate. The final test accuracy, while lower than the one of Epoch 15, at 0.844, is still an acceptable level.

**Task 3:**

Same as task 2, we also trained our model

Comparison with other models:

Here is a comparison table of different models’ accuracy rates in general[cite]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Our Model | AlexNet | VGG-16 | Inception | ResNet-152 |
| Accuracy Score | 0.83 (± 0.19) | 0.86 | 0.901 | 0.941 | 0.963 |

We also tried running a pre-trained model from ResNet-50, the results gathered were 0.94 for task 1.

Overall, our team’s model was able to produce results passing the acceptable threshold, accurately classifying different furniture in task 1 and able to recommend the correct furniture type.

# 5. Conclusion:

In conclusion, our team was proud to be able to create an on-par image classification model for our first major machine learning task. Recognizing our shortcomings on the research and development of our model compared to existing models, we aim to take this as a valuable lesson to help ourselves learn and improve further when we encounter similar problems in the future.

# 6. References:

[1] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review," in *Emerging Technology in Modelling and Graphics*, J. Mandal and D. Bhattacharya, Eds. Advances in Intelligent Systems and Computing, vol. 937, Singapore: Springer, 2020. [Online]. Available: <https://doi.org/10.1007/978-981-13-7403-6_11.>

[2] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of Image Classification Algorithms Based on Convolutional Neural Networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021. [Online]. Available: <https://doi.org/10.3390/rs13224712>.

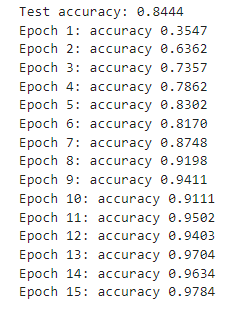
[3] S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith and N. Kumar, "Image Classification using SVM and CNN," 2020 International Conference on Computer Science, *Engineering and Applications (ICCSEA)*, Gunupur, India, 2020, pp. 1-5, doi: [10.1109/ICCSEA49143.2020.9132851](https://ieeexplore.ieee.org/document/9132851).

[4] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Basics of Supervised Deep Learning," in Advances in Deep Learning, vol. 57, Studies in Big Data, Singapore: Springer, 2020. [Online]. Available: <https://doi.org/10.1007/978-981-13-6794-6_2>.

# 7. Appendices:



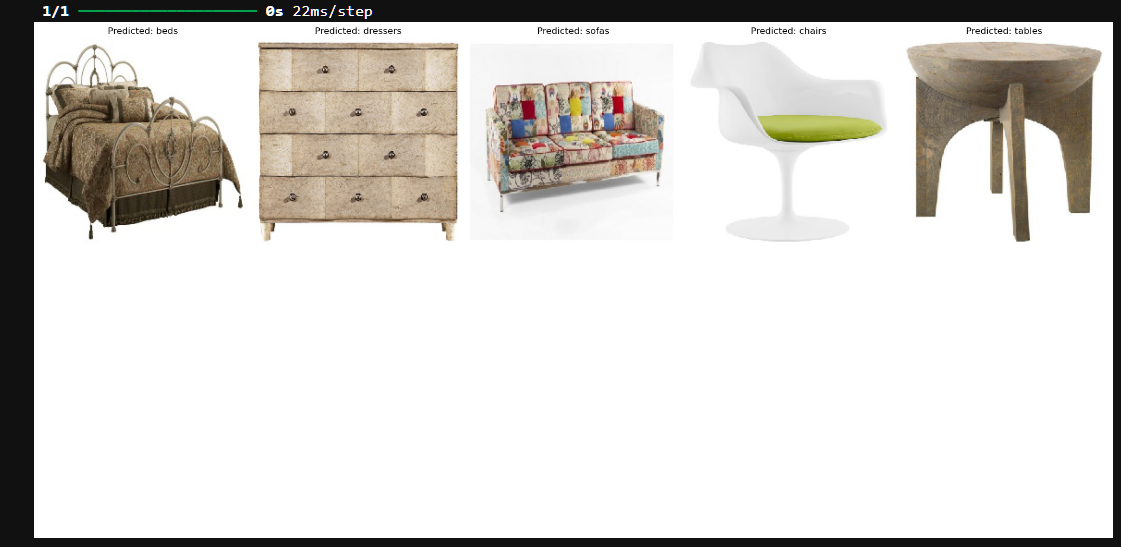
Appendix 1: Task 1 training results.

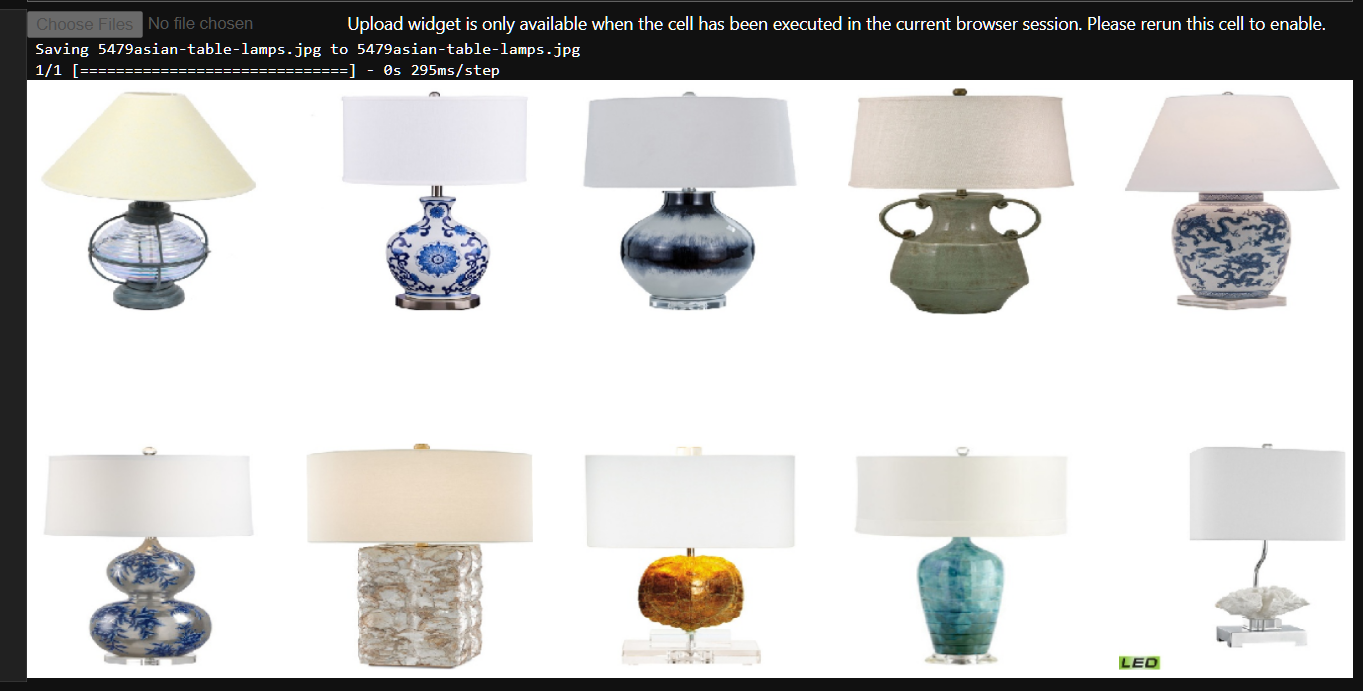


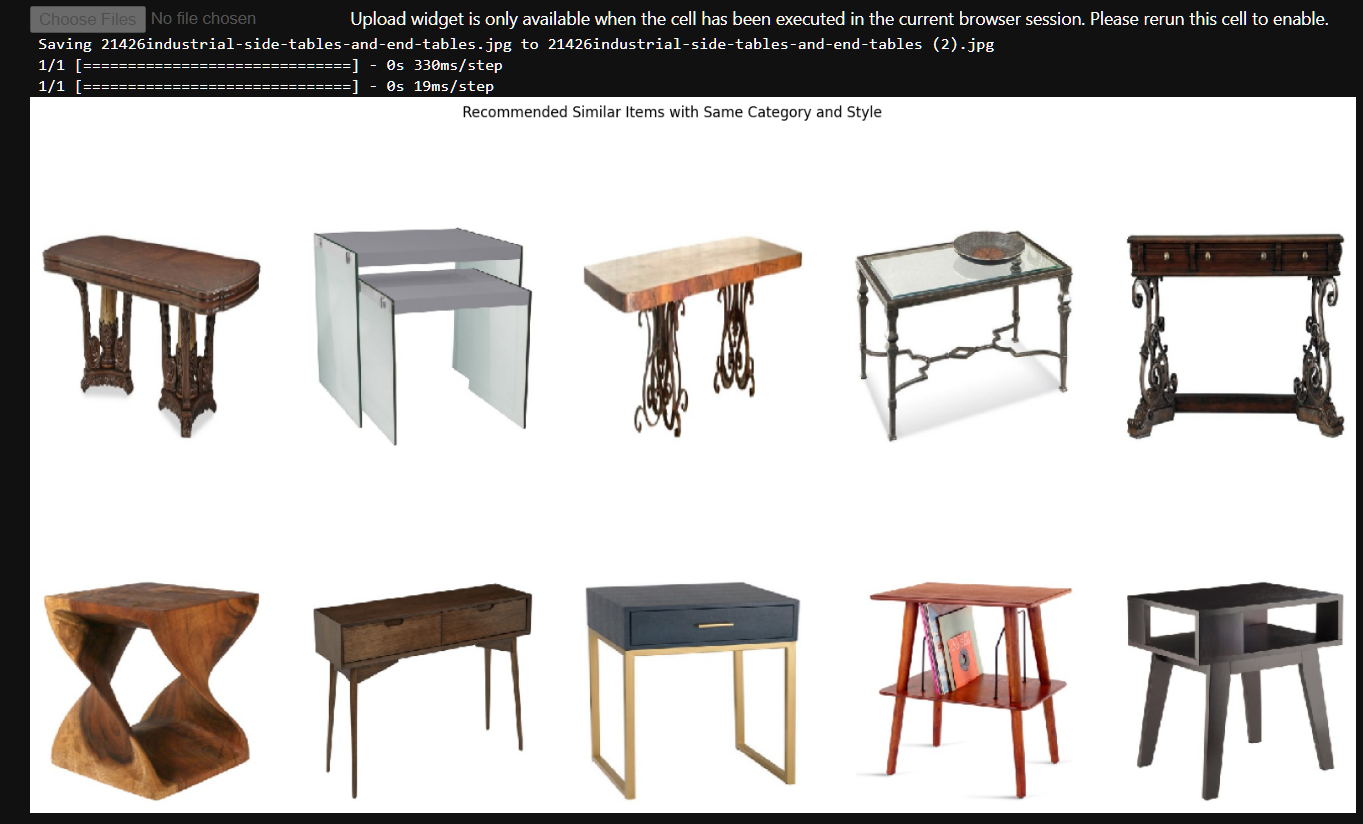
Appendix 2: Task 2 training results.

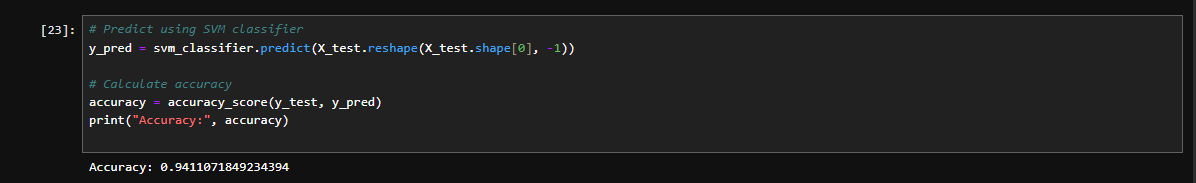


Appendix 3: Task 3 training results.

Appendix 4: Task 1 classification results.

Appendix 5: Task 2 recommendation results

Appendix 6: Task 3 recommendation results.

Appendix 7: Pre-trained model (ResNet-50) accuracy rate for Task 1.