CP217 PROJECT 1: ARE YOU IN A SAFE BUILDING?

Team Name: TensorTitans (Rank - 9) [Final Private: 0.573, Max Private: 0.585]
Team Members: Rajarshi Bhattacharjee, Sandeep Sahu, Santosh Kumar

1 Introduction

The goal of this project is to develop machine learning models to classify Google Street View images of buildings into five classes: Steel, Concrete, Masonry, Wooden, and Steel with panel buildings. The training and testing data contain monochrome images of size 300x400. The class distributions in the training dataset are as follows: A:299, B:362, C:731, D:914, S:210.

2 Data Preprocessing

Visual inspection was performed by directly looking at the images in the 'Train_Data' folder and identifying anomalies. We noted filenames of obviously misleading and noisy images, which were copied to the 'Delete_images' folder. During the preparation of the training data, the program automatically ignores the images that are in that folder.

The criteria for the files to be deleted were: (1) Contains no building; (2) Dark and too noisy; (3) Obviously miscategorized: (i) Masonry violating (no soft story or large openings in the bottom part, presence of decorative patterns); (ii) Steel with panel buildings (violating: no more than 2 floors criterion); (iii) Occluded by trees or vehicles.

Two types of data were prepared for training: one without excluding the 'Delete_images' folder files, and another after excluding them from training. Our preprocessing also involved removing irrelevant/redundant portions of the image, such as the 'Google' text appearing at the bottom of many images. For each image, the bottom 'Google' text (20 pixels) was cropped, and the images were resized to 224x224 (the standard resolution for pretrained models on ImageNet).

3 Methodology and Results

Initially, we tried using classical models: Directly flattened (224x224) images were used with XGBoost and SVM (with a linear kernel). Due to the high computational complexity of SVM ($O(n^2)$-$O(n^3)$)[1], XGBoost and SVM were used. XGBoost was manually tuned based on cross-validation. The XGBoost parameters were: Estimators = 500, Depth = 1-2, Early stopping rounds = 10.

For direct flattened images (224x224), 5-fold cross-validation was performed on XGBoost (achieving 0.404 accuracy) but not on SVM (which achieved 0.33 accuracy). For Canny-filtered images, XGBoost achieved 0.411 accuracy, and SVM achieved 0.417. The reason for the low accuracy is that flattening an image removes the relationship with the neighboring pixel. The increase in SVM accuracy with Canny edge detection is likely due to edges providing important information for classifying clearly defined geometries such as buildings. Sobel filter was also tried without much experimentation, but it yielded worse results. This analysis and the use of filters are not exhaustive.

To utilize information from neighboring pixels and extract patterns, we moved towards convolution-based models. The idea was to extract features using a pretrained CNN, clip off the last fully connected layer, and use SVM or XGBoost instead. We chose ResNet-50 with preloaded ImageNet weights for feature extraction, reducing the number of features from 224x224 = 50,176 to 2,048. Using these features, both XGBoost and SVM (kernel: 'rbf') were trained. Five-fold cross-validation showed that Convo-SVM achieved 0.62 accuracy, and Convo-XGB achieved 0.60 accuracy.

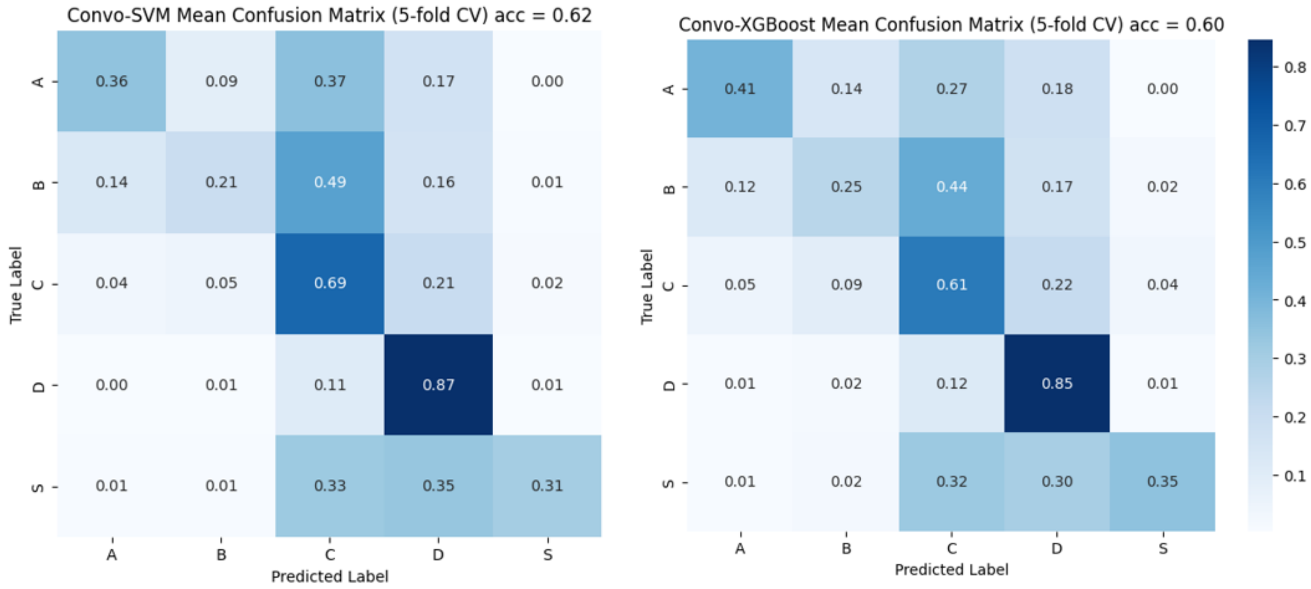

Figure 1: Convolutional SVM and XGBoost Mean Confusion Matrices (5-fold CV)

MODELS	**LOCAL-ACCURACY**
Direct XGB (224x224 features)	0.404 (5-fold)
Direct SVM (224x224 features)	0.330
Canny XGB (224x224 features)	0.411 (5-fold)
Canny SVM (224x224 features)	0.417
Convo-XGB (2048 features)	0.599 (5-fold)
Convo-SVM (2048 features)	0.622 (5-fold)
ResNet-50	0.62 - 0.68
ResNet-50 (Max-Voting-7 Member)	0.7

Table 1: Model Accuracy Comparison

The model achieved more than 0.7 accuracy on classes C and D because the number of images present for these classes was higher compared to the other classes. Classes A, B, and S had significantly lower accuracy, likely due to the smaller number of images in these classes. These classes also had ambiguity and significant overlap.

We experimented with pretrained models such as ResNet-50, ResNet-18, and ResNet-34. The accuracy for ResNet-18 and ResNet-34 was limited to less than 0.6 for unfreezing the last layers. ResNet-101 was very prone to overfitting when more layers were unfrozen. ResNet-50 pretrained on ImageNet stood in the middle ground, achieving a validation accuracy of around 0.65. Using SMOTE (which interpolates and generates new synthetic data), the accuracy of class B increased but at the cost of class A. In our case, SMOTE introduced a tradeoff between oversampling and accuracy, which was noted in several scenarios. It also introduced distortions and artifacts, as can be seen in Figure 2. Figure 3 is the normalized confusion matrix for ResNet-50 and SMOTE.

To combine the benefits of all the models, we used a max voting classifier. This typically performs better than individual models, as expected this performed better for all classes.

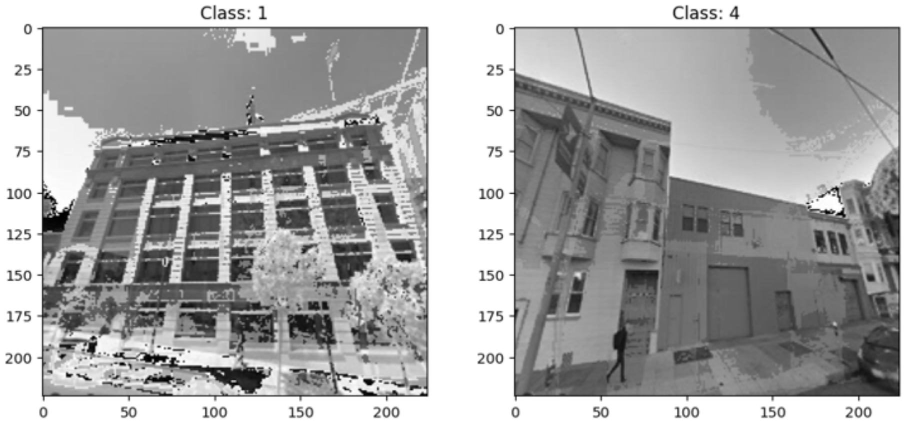

Figure 2: SMOTE Artifacts

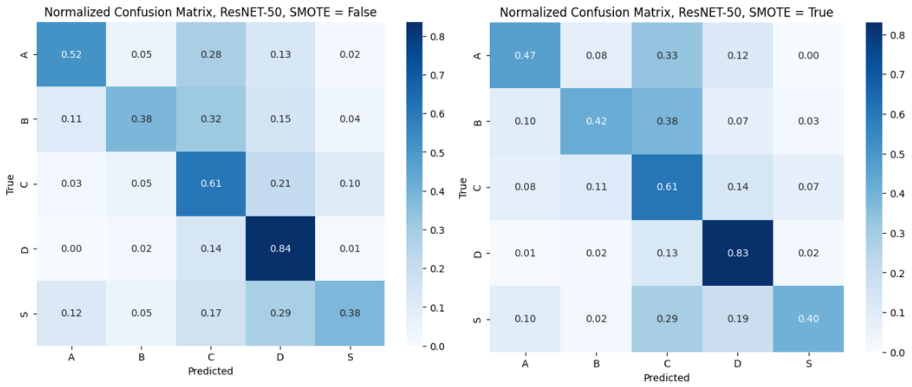

Figure 3: Normalized Confusion Matrix: ResNet-50: without and with SMOTE

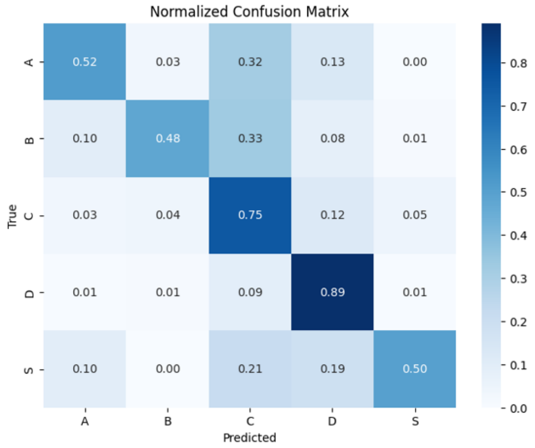

Figure 4: Final 7 Member Max Voting Classifier

4 Conclusion and Future Work

From this project/assignment we were able to understand and demonstrate that the effectiveness of classical Models like SVM increases when they are able to train on extracted/relevant features. This needs further exploration from our side. Instead of just using the classic max voting ensemble we also generated ensembles by averaging softmax-probability outputs of multiple models. Because of the data being low-quantity and noisy and models showing high variance a stacking classifier that takes output of multiple models and makes a more sophisticated ensemble is also something we thought of implementing.

5 References

1. https://www.thekerneltrip.com/machine/learning/computational-complexity-learning-algo

2. Ren, X., Guo, H., Li, S., Wang, S., Li, J. (2017). A Novel Image Classification Method with CNN-XGBoost Model. In: Kraetzer, C., Shi, YQ., Dittmann, J., Kim, H. (eds) *Digital Forensics and Watermarking. IWDW 2017. Lecture Notes in Computer Science*, vol 10431. Springer, Cham. https://doi.org/10.1007/978-3-319-64185-0_28

3. Han, L., Yang, G., Yang, X., Song, X., Xu, B., Li, Z., Wu, J., Yang, H., Wu, J. (2022). An explainable XGBoost model improved by SMOTE-ENN technique for maize lodging detection based on multi-source unmanned aerial vehicle images. *Computers and Electronics in Agriculture*, Volume 194, 106804. https://doi.org/10.1016/j.compag.2022.106804

4. Soad Almabdy, Deep Convolutional Neural Network-Based Approaches for Face Recognition. https://doi.org/10.3390/app9204397

5. Abien Fred Agarap, An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. https://arxiv.org/abs/1712.03541

6. A. Mahajan and S. Chaudhary, ”Categorical Image Classification Based On Representational Deep Network (RESNET),” 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 327-330, doi: 10.1109/ICECA.2019.8 https://ieeexplore.ieee.org/document/8822133

7. Building instance classification using street view images, Jian Kang, Marco Körner, Yuanyuan Wang, Hannes Taubenböck, Xiao Xiang Zhu. https://doi.org/10.1016/j.isprsjprs.2018.02.006