Food Desert CNN : Week 14 Final Milestone

Team Members: Prerna Singh (prerna_singh@berkeley.edu), Yu-Sheng Lee (yushenglee@berkeley.edu), Yiwen Hou (yiwen_hou@berkeley.edu), Kandace Webber (kandywebberlove@berkeley.edu)

1. Abstract

Food deserts are areas with limited access to affordable and nutritious food. They are often identified through static, incomplete datasets with as much as 10 years between updates through census data. In this project, we explore whether satellite imagery and models can supplement or improve the classification of these food deserts. Our approach combines USDA food access labels, census tract shapefiles, and Google Maps satellite images to train multiple machine learning models, including a convolutional neural network (CNN). We test three approaches: a random forest as a baseline model, an image-only CNN, and a hybrid CNN + metadata model. The CNN+Metadata model has the highest accuracy, just above the Random Forest Model and finally the image-only CNN model was the least accurate. We found that census data is necessary in order to enhance the predictive performance as socioeconomic features such as poverty rate, no-vehicle households, and urban/rural setting provide richer contexts for visual patterns from imagery data. Although satellite images alone do not provide sufficient predictive power, in scenarios where census data are outdated or unavailable, the hybrid CNN model can be a promising method.

2. Introduction

A food desert is a geographic area where residents lack convenient access to affordable, nutritious food usually due to the absence of grocery stores within a reasonable distance. These areas are often defined at the census tract level using income and distance-based criteria by the USDA. However, existing classifications rely on outdated census data and require extensive manual data collection, which limits responsiveness and coverage, especially in rural or low-income areas. In this project, we investigate whether visual patterns in satellite imagery can serve as predictors of food deserts. We hypothesize that overhead imagery of the environment contains latent features that correlate with food access. These may include road density, green space, commercial clusters, or housing patterns. The input to our algorithm is satellite imagery (RGB, 400x400 pixels) centered on the centroid of each census tract in the southeastern United States. We also incorporate metadata features such as tract type and state. We use a random forest model as a baseline, a convolutional neural network (CNN) to classify images, and a combined model that integrates image and metadata features to predict binary food desert status.

3. Related Work

The first article we reviewed was Combining Satellite Imagery and Machine Learning to Predict Poverty[1]. This article attempted to identify areas of low-income and poverty in five African countries by using a CNN to process satellite images. One strength of their model was that they used raw nightlight data to first train the CNN to predict nighttime light intensity from daytime satellite images. This helped them overcome sparse data that they had available to them. Their approach gave us ideas on how we could solve our problem. Compared to our work, their method is mainly focused on rural expanses while ours has a mix of tracts. Their additional feature was effective which we were hopeful would be similar to combining our metadata and our CNN. Both projects demonstrate the potential of CNNs for social good when conventional data sources are limited or outdated.

The second article we reviewed discussed the different ways that organizations have attempted to identify food deserts[2]. The authors also challenge the concept of “food deserts” and argue that geographic proximity to grocery stores is a poor proxy for actual food insecurity. Their research suggests that the assumptions that people shop at the nearest store, overlook community-based food coops, and don’t account for online food delivery or individual mobility strategies like trip chaining. Their central argument is that financial insecurity, not geographic access alone, is the true driver of food insecurity. Compared to our work, which classifies tracts using satellite imagery and census-defined food desert labels, their critique suggests our labels may not reflect modern food dynamics. However, our CNN-based visual model aims to offer a scalable, real-time supplement to these traditional mappings. While George and Tomer focus on the structural and economic causes of food insecurity, our model could still contribute as a diagnostic tool, especially in areas where digital purchasing and delivery are not yet viable.

4. Dataset

Data Source

Our project integrates 3 key datasets to create a tract-level classification problem:

1. **USDA Food Access Research Atlas**
   This dataset provides binary food desert classifications at the census tract level, based on criteria combining low income and low access to grocery stores. We define our target variable IsFoodDesert using the USDA’s urban/rural composite metric: tracts where LILA_Urban1_Rural10 = 1 are labeled as food deserts.

---

[1] Jean et al, 2016 - Reference Page
[2] George and Tomer, 2021 - Reference Page

2. [U.S. Census TIGER Shapefiles]
   These shapefiles contain the geographic boundaries and centroid coordinates for every census tract in the United States. We use the latitude and longitude from these shapefiles to extract image-centered locations.
3. **Google Maps Static API (Satellite Imagery)**
   Using the tract centroids, we retrieved 400x400 pixel RGB satellite images centered on each tract. These serve as the visual input to our CNN. Each image corresponds to one census tract and inherits the USDA label.

There are originally 72,531 records in the atlas dataset, where each record represents one unique census tract. After joining the datasets and removing tracts with missing or invalid coordinates, we focused our analysis on 8 southeastern states: Tennessee, Georgia, Alabama, Kentucky, South Carolina, Louisiana, Arkansas, and Mississippi. This resulted in 7,121 census tracts, each with a binary food desert label and an associated satellite image.

Data Preprocessing

	1. Random Forest	2. CNN (Image)	3. CNN (Image + Metadata)
Original Atlas Data	72,531records		
After filtering for 8 southern states	7,121 records remained		
After removing missing values for LAPop_1Mile and LAPop_10Miles	N/A	1,364 records remained	
Train/Validation/Test Split	(0.8, 0.2) => (5696, 1425)	(0.6, 0.2, 0.2) => (818, 273, 273)	
One-Hot Encoded	StateName and TractType converted to binary indicator for model interpretability		
Standardization for numeric features	No need for random forest	Standard scale all population and income measure variables	
Image Pre-Processing	No need for random forest	Convert to 256x256 RGB image size	

We derived our target variable, *IsFoodDesert*, from the USDA-defined indicator *LILA_Urban1_Rural10*=1, which identifies census tracts as food deserts based on both low income and low access criteria—being more than 1 mile (urban) or 10 miles (rural) from a supermarket. Selected features can broadly represent the **socioeconomic status** of each tract, including *IsUrban, PovertyRate, and LowVehicleAccess*..etc., along with population-related variables such as *GroupQuartersPopulation, WhitePopulation,* and *NHPIPopulation*..etc. We included *LAPop_1Mile a*nd *LAPop_10Miles* in theCNN model to reflect the population sparsity in both urban and rural settings. These two features were used only in the CNN models and excluded from the random forest model. As a result, the random forest model retained a larger sample size of **7,121**. However, due to the high proportion (~80%) of missing data in *LAPop_1Mile* and *LAPop_10Miles,* imputation was deemed inappropriate, limiting the CNN models to a smaller dataset of **1,364** tracts. Since training a random forest model does not require a validation set, we utilized 80% of data as the training set. For both CNNs, a 20% validation split is required to monitor performance during training.

EDA

Our EDA revealed several insights into our data:

1. Class imbalance: The distribution of food desert and non-food desert tracts is extremely imbalanced, with 78% of non-food deserts.[3] This may lead to the model not learning enough information about food deserts and recognizing more as non-food deserts. Additionally, most census tracts are urban, but most food deserts are disproportionately in urban tracts.[4] This helped inform us that we would need to use a weighted approach.
2. Geographic clustering: Food deserts were primarily concentrated in urban areas and the outskirts of cities[5]
3. Correlation matrix: IsFoodDesert label strongly correlates with low income and rural areas, confirming how the label was defined[6]

This analysis supported our decision for a binary classification approach. They also influenced our decision to use metadata to improve generalizations across geographies.

Data Challenges

---

[3] Appendix Figure 3
[4] Appendix Figure 1
[5] Appendix Figure 2
[6] Appendix Figure 6

To address class imbalance, we brought class weightings into the model fitting process, which informs Tensorflow to allocate more attention to the under-represented class by adjusting the loss function accordingly. Specifically, this can be achieved by passing a dictionary that contains the distribution of both classes into the class_weight parameter. Another challenge we faced was the time discrepancy between the images and the metadata. The images are retrieved real-time from Google Maps API, but the census data is only updated once every 10 years. The most recent available version is published in 2019. For future improvements, we should explore ways to obtain historical Google imagery that aligns with the metadata timeframe.

5. Methods

Model 1 Baseline: Random Forest (Census Data only)

For our baseline model, we conducted a random forest which was trained only on metadata from the USDA Food Atlas data. The majority of tract-level attributes were included, excluding different versions of label features, state identifiers, and GIS mapping variables. We chose a Random Forest model to combat overfitting and to identify important features. The metadata originally contained 152 features. After cleaning and processing the labeling data, the feature set was reduced to 20. Using the Random Forest algorithm, we identified the most important features through entropy-based information gain and randomized iteration. The model produced helps classify observations based on our label definitions.

Model 2: CNN Satellite Images Only

Convolutional Neural Networks work by applying convolutional layers to detect edges, shapes, etc in an image. The layers are then pooled to reduce computational complexity and downsample feature maps. These feature maps are then connected in a layer and the model makes a classification prediction. We chose to start with a CNN image only model to determine whether satellite images provide enough information to make classification decisions.

To test our hypothesis of whether a model can better predict food deserts than outdated Census data, we trained a CNN using only 256x256 RGB satellite images as input. We normalized the images and passed through several convolutional, max pooling, and dense layers to capture visual patterns indicative of infrastructure, density, or green space.

We experimented with different hyperparameters, changed the number of convolutional layers, filter sizes, dropout rates, and learning rates. We used TensorFlow to process the model. To reduce overfitting and improve generalization, we used early stopping and manually minimizing the layers used by observing several model evaluation metrics.

Model 3: CNN + USDA Metadata

Our final model integrates USDA metadata and the image classification from a CNN. We selected this approach to improve generalizability. Given that there can be many variations among states, we selected features from the original dataset that could improve classification. While the standalone CNN model captures only visual patterns associated with food deserts, this hybrid approach incorporates additional contextual information like state, urban/rural classification enabling the model to account for geographic and demographic variation.

We selected metadata features based on both their predictive value and practical availability. We added low-access population measures (within 1-mile and 10-mile radiuses) to complement the image features. These variables are easier to obtain than full census demographic profiles where data may be unreliable. By limiting our metadata to more readily available features, we aimed to build a model that would be more accurate, and more useful to researchers or policymakers working in data-sparse regions.

Model training followed the same tuning strategies as our standalone CNN in addition to testing out different features. These three models represent increasing complexity and access to data. Our hypothesis was that combining visual and non-visual features would yield the most accurate and generalizable predictions of food deserts.

6. Experiments, Results, and Discussion

As mentioned above, since the majority class, non-food desert, takes up 78% percent of total data points, any model's performance should have higher than 78% accuracy to be considered better than random guessing a non-food desert. "The simplest model that performs well is often the best choice" is the principle that we follow through to fine-tune our models.

Hyperparameter tuning strategies and validation methods for Random Forest

The random forest model is our baseline model.[7] Generally speaking, random forest has fewer hyperparameters compared to other machine learning models - especially compared to models like neural networks or gradient boosting machines.

- We set bootstrap=False to let each tree train on the entire training set instead of a random sample with replacement.
- Starting with max_depth=20 and n_estimators=20, we limit the maximum depth and number of estimators to the minimum when keeping the testing accuracy the same to prevent overfitting. Table 2 shows that under the same accuracy, we prefer the model with minimum hyperparameters (the 2nd one)[8]. Although the training accuracy of the 1st random forest is 100%, it has a bigger gap (16%) between the test accuracy, indicating potential overfitting.

---

[7] Appendix Figure 7

[8] Appendix Table 2

- We set the splitting criterion as “entropy” instead of the default gini. The former measures the impurity or disorder in the data and comes from information theory. It's used to decide how to split the data in decision trees. If all samples at a node belong to the same class, then entropy = 0.
- Since training a random forest model does not require a validation set, we can utilize 80% data to boost the training power. The remaining 20% of the data is used to evaluate the model performance.

Hyperparameter tuning strategies and validation methods for CNNs

In our CNN model experiments, we first started a CNN with three convolutional blocks on 256x256 RGB satellite images only. Each block included a ReLu activation and max-pooling layer. The final layers included a flattened output, a dense ReLu-activated layer, and a sigmoid classifier for binary output.[9] The model used the Adam optimizer with a learning rate of 0.03%, binary crossentropy loss, and early stopping on validation accuracy with a patience of 5 epochs. After we got a baseline understanding of how image-only CNN works, we tested the CNN + Metadata Fusion Model with various combinations of tabular features from the census data and selected a subset that offered a good tradeoff between generalizability and performance. The hybrid model has an extra branch to consume census metadata before the concatenating layer that combines both image and tabular results.[10]

Both models share similar hyper-tuning and validation strategies. We used selective manual tuning and leveraged Keras Tuner for early experiments which focused on:

- Learning rate: we manually tested and found 0.0003 to be stable for the Adam optimizer
- Input image size: we reduced the dimensions from 400x400 to 256x256 (or smaller) to improve training speed
- Batch size: Initially we used batch size of 32 but later increased to 64 to improve GPU performance
- Early stopping: we used early stopping with patience = 5 on validation to prevent overfitting
- Class Balancing via Weights: To mitigate the class imbalance, we calculated weights and applied them during training. This helped improve recall for underrepresented food deserts tracts.
- Threshold Tuning: We tested threshold below 0.5 to increase food desert detection (not successful)

To evaluate model performance, we used a train/validation/test split strategy. The 60% training set was used to fit model weights, and the 20% validation set was used for model selection, early stopping, and tuning decisions. This means that the model would stop training if validation accuracy did not improve for 5 consecutive epochs, and the best weights were restored at the end. The 20% test set was held out only for final evaluation to estimate real-world performance. This helped avoid overfitting.

Evaluation metrics and Subgroup performance evaluations

We used the following evaluation metrics: accuracy (training, validation, and test), confusion matrices, precision, recall, and F1-scores. Accuracy was primarily used during training and validation. We also looked at other metrics because it can be misleading with imbalanced data. The confusion matrix helped us understand how the two classifications broke down. It helped us understand whether the model was overpredicting or underpredicting food deserts. Precision and recall helped us understand how many predicted food deserts were correct vs how many the model identified. Finally, the F1-score helped us understand the harmonic mean of precision and recall.

According to the model performance table in Table 1[11] in the Appendix section, the random forest model’s test accuracy is 82%. While both precision and recall are higher for the non-food desert group compared to the food desert group, a closer look reveals that the non-food desert group achieves significantly better results in both metrics. For the food desert class, we achieved a 64% precision rate and a 60% recall rate. However, since our priority is to accurately identify food deserts, we continued exploring other models, such as CNN and integrated CNN models, to improve performance.

The image-only CNN model has a 0% precision, 0% recall, and 0% F1 score, indicating that it can never correctly identify a food desert. The overall accuracy is only 78%, which is the same as the proportion of non-food desert tracts in the whole data set. Additionally, the misclassified images are all false negatives (true food deserts identified as non-food deserts). They contain both high and low population density tracts.[12] In other words, image-only CNN collapses to predict only the majority class (non-food desert) for every example, regardless of whether extra layers are added to the model.

In contrast, image + metadata CNN has a 58% precision, a 78% recall rate, and a 67% F1 score. The overall accuracy is 83%, which shows a certain degree of ability to distinguish between a food desert and a non-food desert tract. When looking at the misclassified images for the image + metadata model, we see that all images have a big portion of green space.[13] The left-hand side images are all false positives, and the right-hand side are all false negatives.

Based on the overall accuracy rate, we chose the hybrid image and metadata CNN as our final model. The baseline random forest achieved the accuracy at 82%, due to the strong alignment between its input features and the definition of food desert by the USDA. The hybrid CNN outperforms image-only CNN because it contains extra information to tell if socio-economic status like if

[9] Appendix Figure 4
[10] Appendix Figure 5
[11] Appendix Table 3
[12] Appendix Figure 8
[13] Appendix Figure 9

these tracts are residential areas or simply farmlands, which highlights the importance of socioeconomic features from the metadata.

7. Conclusion

In this project, we explored whether satellite imagery and CNNs could be used to predict food desert status across census tracts. Using USDA-defined food desert labels and recent Google Maps satellite imagery for eight southeastern US states, we built three core models: a random forest for only metadata features, a CNN trained on images alone, and a CNN with metadata.

The random forest model technically relies entirely on government metadata — the same data that informs the USDA labels. This raises a key issue: Census data is only collected every 10 years, and in the current political climate, its accuracy, frequency, and public availability are uncertain. In contrast, CNNs using satellite imagery can offer a scalable and more frequently updatable alternative. Our findings suggest that while metadata models are more accurate under current conditions, image-based models may offer critical value in future scenarios where census data is outdated, incomplete, or unavailable. Therefore, we choose the hybrid CNN as our final model.

In future iterations, we recommend expanding the geographic scope of the dataset to include a broader range of regions for better generalizability. We also see strong potential in integrating additional geospatial signals — such as the density of grocery stores or land use patterns — using APIs like Google Places. These additions could further improve model accuracy and provide nonprofits and policymakers with powerful tools for identifying at-risk communities in real time, independent of decennial census cycles.

8. Contributions

Each of us contributed to every stage of the paper to ensure that everyone had the opportunity to learn all aspects of the project. This approach also allowed us to cross-reference our results more effectively.

Kandy	Prerna	Yu-Sheng	Yiwen
EDA, logistic regression, CNNs modeling, and writing up reports and presentations. Webber 207 ML: Food Desert EDA Webber_PCA_Logistic_Regression Proposal Progress Milestone Final Report	EDA, logistic regression, CNNs modeling, and writing up reports and presentations. Prerna's_eda Prerna-PCA_Logistic_Regression CNN Exploration Proposal Progress Milestone Final Report	EDA, logistic regression, CNNs modeling, random forest, writing up reports and presentations. 01_RandomForest_Yusheng 01_Yusheng_inital_analysis 02_CNN_w_image_Yusheng 02_CNN_w_metadata_Yusheng_1 02_CNN_w_metadata_Yusheng_2 02_CNN_w_metadata_Yusheng_3 02_CNN_w_metadata_Yusheng_4 02_PCA_logistic_Regression Proposal Progress Milestone Final Report	EDA, logistic regression, CNNs modeling, random forest, writing up reports and presentations. CNN_Yiwen CNN+Meta_Yiwen EDA_Yiwen RandomForest_Yiwen Proposal Progress Milestone Final Report

https://github.com/singhprernap/ucb_mids_207_Final_Project_Food_Deserts

References

Jean, N. et al (2016, August 19). [Combining *satellite imagery and machine learning to predict poverty*] | *science*. Science.Org.

Adie Tomer, Caroline George, (2021, August 17). *Beyond "Food Deserts": America needs a new approach to mapping food insecurity.* Brookings.

Appendix

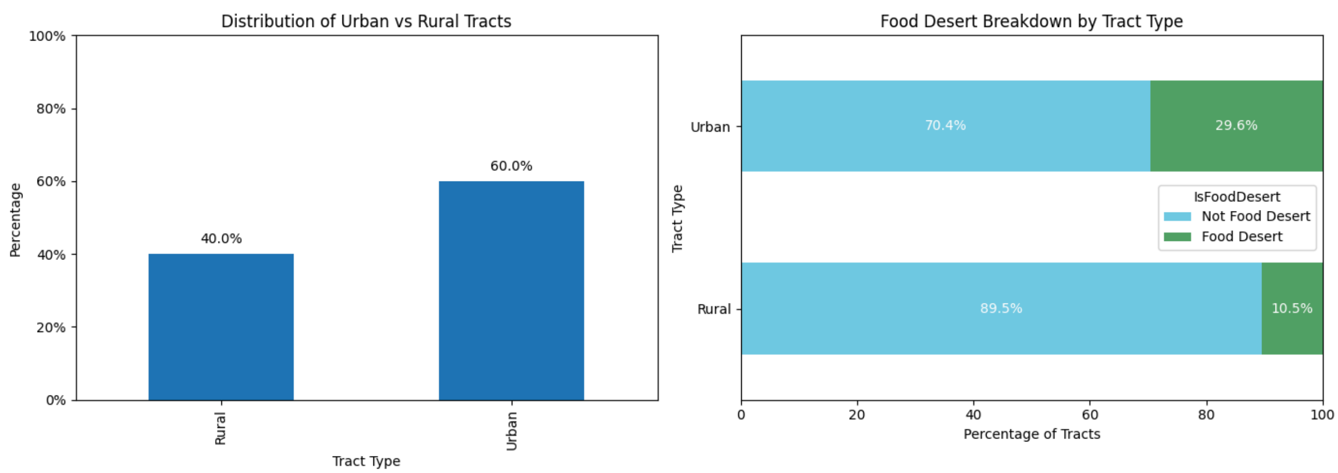

Figure 1: Distribution of Food Deserts by Tract Type

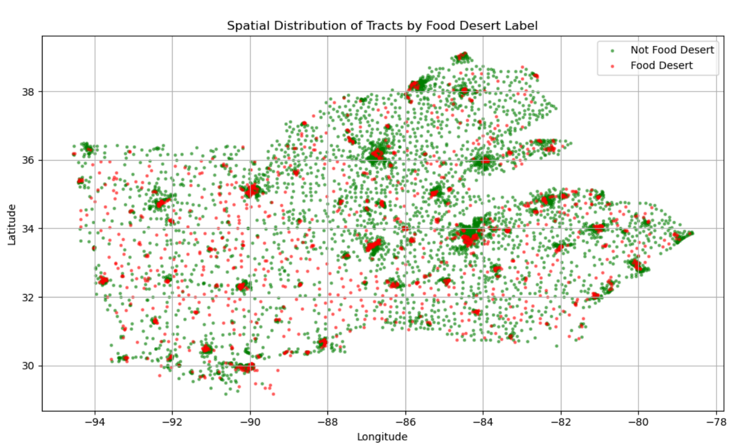

Figure 2: Distribution of Food Deserts and Non-Food Desert

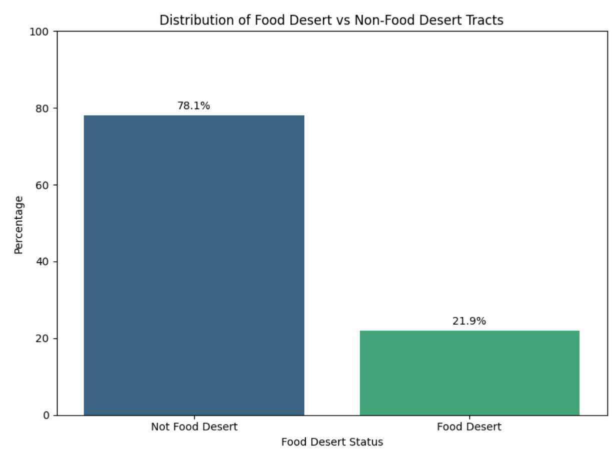

Figure 3: Geographic Spatial Distribution of Food Deserts

Figure 4: Architecture of Model 2 (CNN Image)

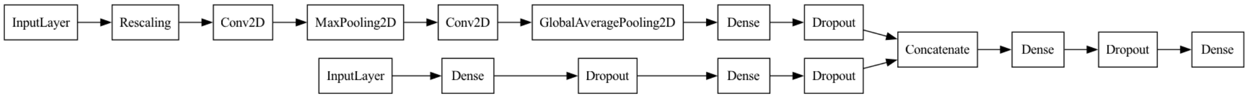

Figure 5: Architecture of Model 3 (CNN Image + metadata)

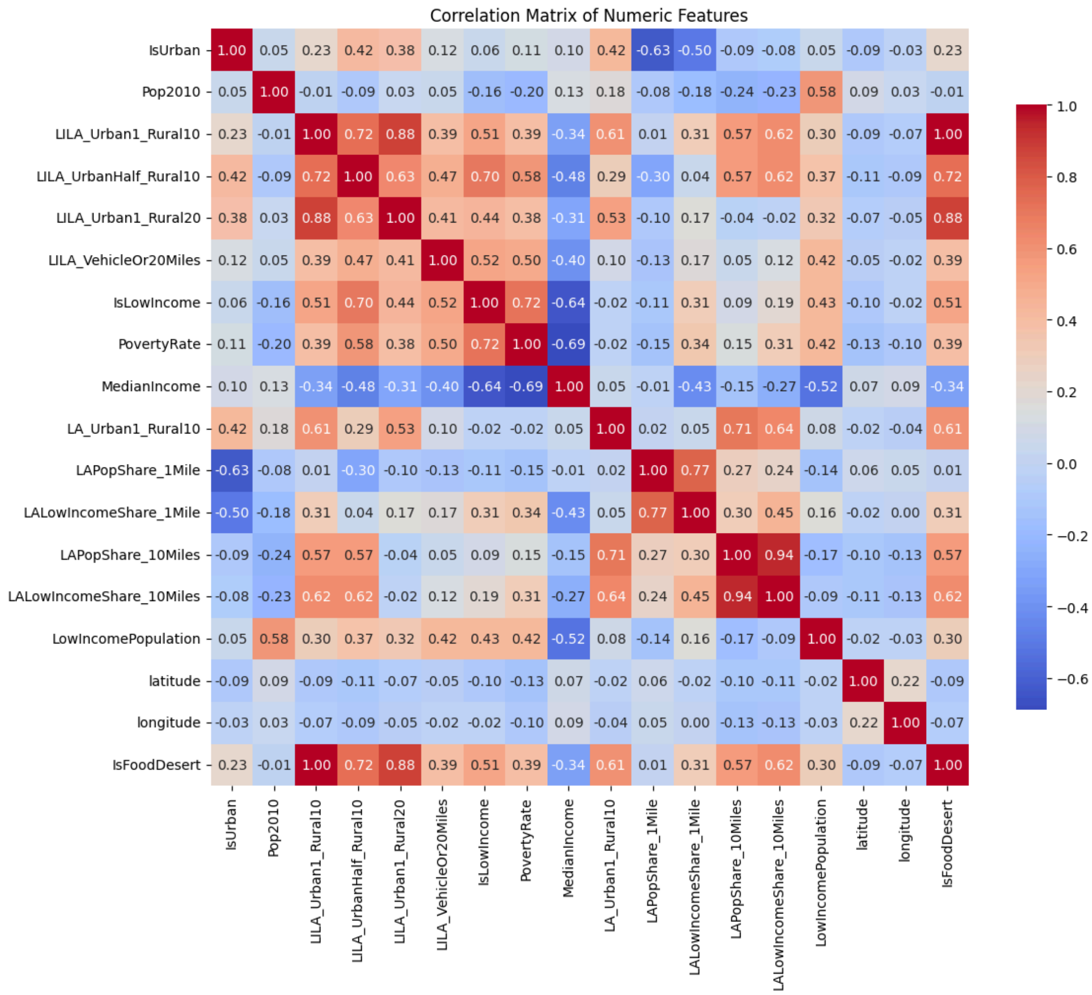

Figure 6: Correlation Matrix of Numeric Features

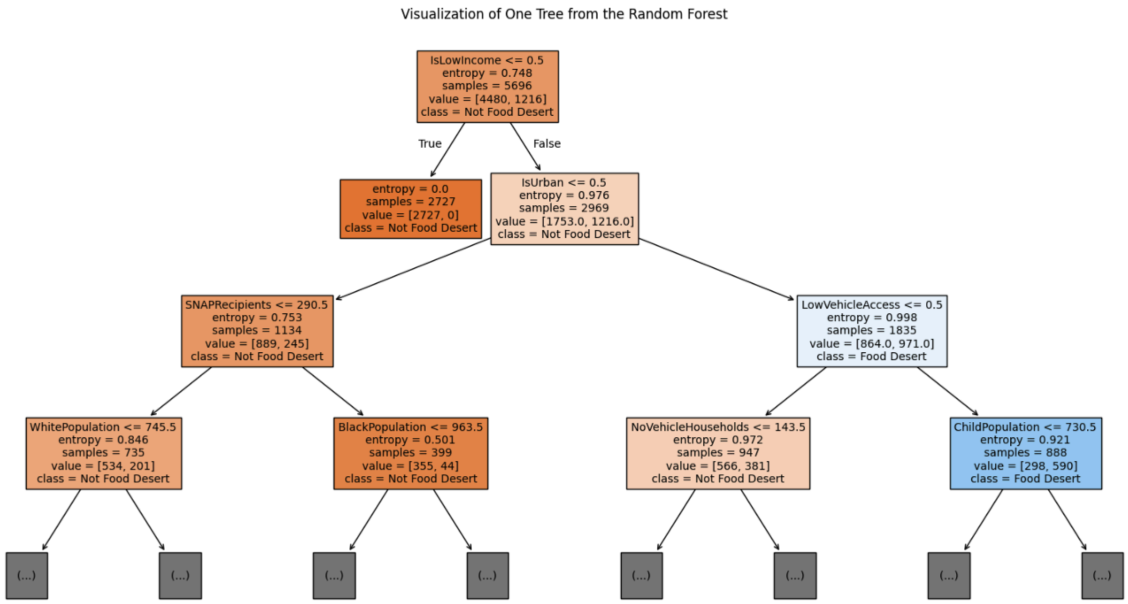

Figure 7: Random Forest Model

Figure 8: Misclassified Images for Model 2 (CNN with satellite image only)

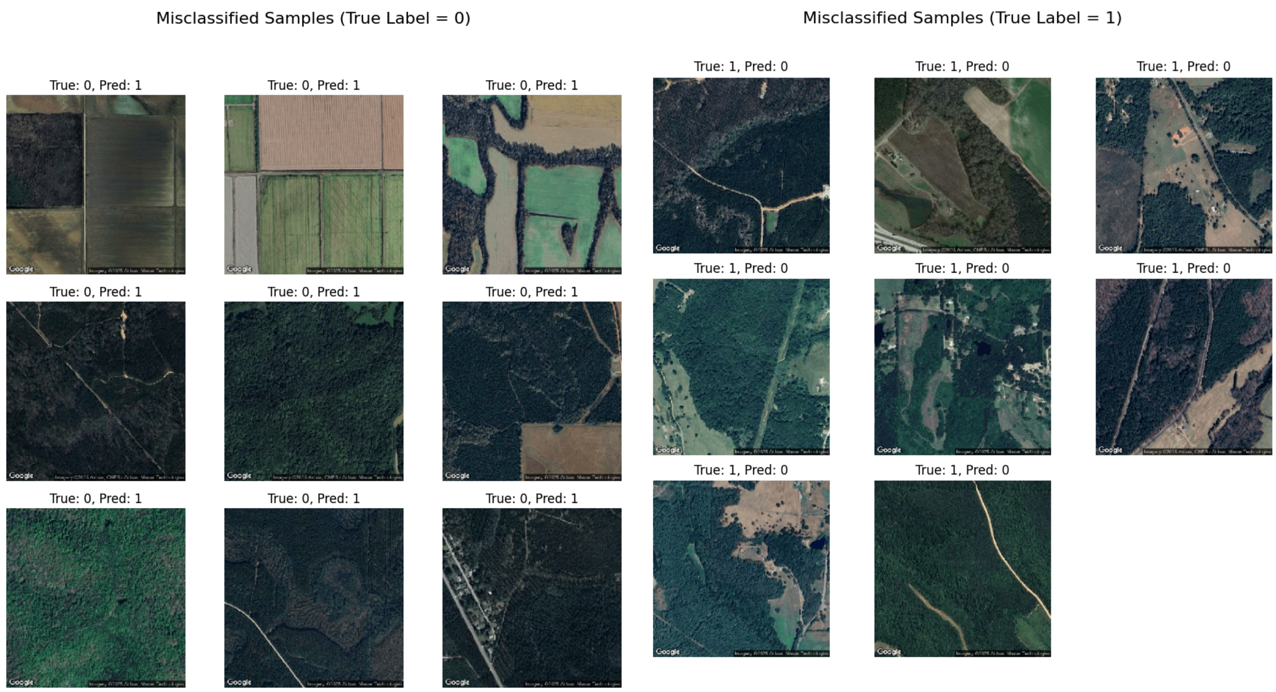

Figure 9: Misclassified Images for Model 3 (CNN with satellite image + metadata)

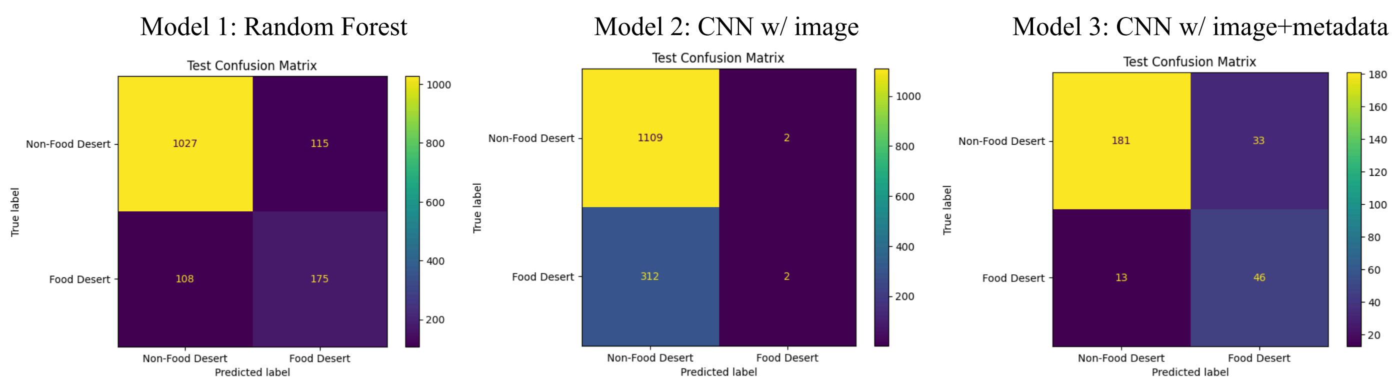

Figure 10: Confusion matrix for 3 models

Table 1. Model Evaluation Metrics

Metrics \ Model	1. Random Forest	2. CNN Images Only	3. CNN Image + Metadata
Test Accuracy	82%	78%	83%
F1 Score	62%	0%	67%
Precision	64%	0%	58%
Recall	60%	0%	78%

Table 2. Random Forest Model Tuning

	Random Forest 1	Random Forest 2 (selected)

n_estimator	20	10
criterion	entropy	entropy
n_jobs	2	2
max_features	None	None
bootstrap	False	False
max_depth	20	8
Training accuracy	100%	89%
Testing accuracy	84%	82%