ITCS 4152/5152: Semester Project

A. Project Proposal

For our semester project, we propose a computer vision application that has the capability of identifying buildings across the UNCC campus. The application will be able to take input media and return the name of the building captured within the given media. Over the course of the semester we will train a model using photos of buildings on campus that are taken from different angles, different times of day, and in different types of weather.

Our target audience will mainly consist of computer science students who want to use the model we will have created in order to develop more software using our building detection as a service within their application. One example of an application that could be derived from our model would be an application that allows students to tour the campus on their own time. By using our building detection product, the student could use their camera on their phone and depending on which building the camera is pointed at, the student could receive detailed information for the given building.

The product will use computer vision techniques for object detection to identify multiple, notable buildings on campus. Multiple sample images of the buildings from different angles and in different lighting scenarios will be fed into the model in order to train it to detect the buildings. The software will utilize convolutional neural networks to detect the buildings and categorize them appropriately.

Our production process is going to be built upon various technologies. Jupyter Notebook is the preferred platform to build the model. To build an effective model, it's important to gather diverse and meaningful data. Sources such as Google Images, UNCC web images, and personal-camera images will supplement our model with quality data. Annotation tools will be applied to the videos and images to feed the ML algorithm labeled-data. Packages like SK learn and ML libraries will be used to process data and provide insight into identifying the buildings. Visualization packages will be used to uncover and understand certain trends of the ML algorithm. For the working process, the management tool, Trello, will be used to monitor progress in an organized manner. In addition, Discord will serve as the primary form of communication among members.

Finally, there will be a demo that showcases the functionality of our product. Given various inputs, including different angles and lighting scenarios, the model will be able to detect which building on campus is being captured in the given media.

Group Member 1

Name: Madison Melton
Undergraduate/Graduate Status: Graduate
Email: mmelto21@uncc.edu

Group Member 2

Name: Rachel Miranda
Undergraduate/Graduate Status: Graduate
Email: rmirand1@uncc.edu

Group Member 3

Name: Prasheeth Venkat
Undergraduate/Graduate Status: Undergraduate
Email: pvenkatk@uncc.edu

Group Member 4

Name: Nathan Williams
Undergraduate/Graduate Status: Undergraduate
Email: nwill111@uncc.edu

Group Member 5

Name: John Taylor
Undergraduate/Graduate Status: Graduate
Email: jtayl247@uncc.edu

https://www.overleaf.com/project/632c7c4e56b3ebfecd1e1dac

B. Research

To prepare for this project we conducted a research phase in order to learn more about our relevant area of Computer Vision. During this phase of the project, we each selected an article related to our topic to review (totaling 5 articles) and also chose 2 questions each to address.

B.1. Article Reviews

What is that Building? An End-to-end System for Building Recognition from Streetside Images

This article describes a system for recognizing buildings using street level images (i.e., an image taken from a bystander standing on the street). This is a significant task when you consider that if a person is taken an image in the downtown area of a big city, they may not be able to capture the entire structure. This would prove to be a slightly different task than image recognition of buildings in more rural areas. To improve accuracy of recognition, these researchers have implemented a feature on Streetside Building Search-Retrieve System (SBSRS) that allows a user to search by image and location. The images stored in SBSRS have associated data that identifies the location of the image which assists with narrowing down possible candidate buildings when searching by image and location. The image database is also populated with semantic information for each building such as business name, open hours, etc. This is very similar to the task we are trying to accomplish as our goal is for a user to be able to take a photo of a building on campus and the software to return the name of the building and other relevant information. In order to handle possible error, the system described in this article has an error threshold of 100 meters. This means if the location is identified to be a certain latitude and longitude, the system will consider that the building could be within 100 meters of this location, in case the user is far away or there is GPS error. The researchers found that their system outperformed the DELF-GLD and DELF-GLD+Location (state-of-the-art models for comparison) on all tested datasets. They attribute this outcome to “adding spatial filtering prunes the search space significantly, improving the average precision in some cases by over thirty percentage points” and “the global descriptors computed by the Siamese network.”

Building Recognition Using Local Oriented Features

For the project that we are working on, we want to be able to provide fast and efficient results when it comes to building recognition. In the paper “Building Recognition Using Local Oriented Features” written by Jing Li and Nigel Allinson, Li and Allinson propose a method for building recognition using local oriented features stating that, compared to other techniques, it is more modular and computationally efficient. At the start of the paper, Li and Allinson provide a background into building recognition and the various uses it has. They then go on to explain some issues that come with building recognition. Things like different angles of photos, objects blocking parts of the building, different lighting conditions, and more are all obstacles that come up when building recognition is attempted. Li and Allison state that they wanted to find a more efficient and simple method of building recognition. They propose a building recognition model called the steerable filtered-based building recognition (SFBR) model which uses local oriented features in representing an image. The SFBR model consists of four different parts. These parts are feature representation, feature pooling, dimensionality reduction, and classification.

Li and Allinson go on to elaborate on the four parts of their model. To have the model work with buildings, a feature representation is needed that is robust when it comes to scale and occlusions. To solve this problem, they decide to use second-order steerable filters at 8 different orientations. This results in sixteen different feature maps that can be used. Moving onto feature pooling, this is done to help the model be more resilient to issues when it comes to shifts in positions and changes in lighting in photos. To have robustness when it comes to image noise, max pooling is used instead of sum pooling. This helps retain useful information while also removing irrelevant noise from the image. In their case, they split their feature maps into 4x4 regions and use max pooling in each region. From there, dimensionality reduction is then done. The feature maps are reduced from a dimension of 256 to a dimension of 39 by using LDA. LDA is a supervised learning algorithm that uses class information to separate examples from different classes and keep examples of the same class together, given a set of labeled training examples. After this, the picture is then classified.

In their experiments, Li and Allinson split their data in half. One half they use for training, and the other half for testing. In their trials, they found that their model would on average give them an accuracy of 94.66%, resulting in higher precision than other models that were tested. In their conclusion, Li and Allinson state that while they got good results from this model, it should not be used to substitute building recognition algorithms. Instead, it should be used as an alternative solution. The main benefits of this model are ease of use and its modularity.

This helps us with our project as we are going to need to have a model that is efficient and easy for us to use. We do not have access to computers with insanely high computing power, therefore we are going to want to use the most computationally efficient method possible.

LIME: Low-Light Image Enhancement via Illumination Map Estimation

One of the problems that is inherent regarding computer vision systems is illumination. Finding details within darker regions of an image is challenging, and Guo, et al. attempted to create a method that would allow for the enhancement of darker images that would retain the already present details while enhancing the details of the features that were darkened.

While the most simple method would be to amplify the entire image, this often leads to over-saturation and loss of detail in the already illuminated portion of the image. The authors pointed out several methods of low light enhancement but also showed how they can be unreliable or how they are methods that don’t actually address the problem of low light. One of the main methods used is called histogram equalization, and while this method is fairly good at extracting detail from dark images, the method is focused on controlling the contrast of the image rather than exploiting real illumination causes. Another method that is often used is gamma correction which works well. However the authors point out that this method focuses on individual pixels rather than looking at neighboring pixels for more information about the image.

In order to address the shortfalls of the typical methods of illumination enhancement, Guo and his team were able to more accurately derive details from images by estimating their illumination maps. This is done by finding the max intensity of each pixel in the R, G, and B channels. The structure of the illumination is then exploited, and the illumination map is refined. Finally an Augmented Lagrangian Multiplier algorithm is performed in order to refine the image. Their method of low-light enhancement was found to be nearly twice as fast as a typical histogram equalization implementation and just as fast as other typical methods. The caveat being that Guo's teams' images turned out to have higher detail.

This article helps to identify possible methods of low-light image enhancement, while also showing their potential downfalls and performance while also providing a method that matches the speed of other widely used methods while providing results with higher detail.

Building a recognition system based on deep learning

To tackle the complex task of building recognition, it's crucial to be aware of the tools and algorithms that are viable. In our case, our data which is likely to be hand-captured by our group around campus will result in a relatively small data sample. An article regarding building recognition, ”Building recognition system based on deep learning”, by Pavol Bezak discusses the methodology used for a viable model with a small dataset and limited memory computing.

Due to real-life conditions, building recognition is often a complex problem. The varying lighting, angle, and resolution are often major factors contributing to the complexity of building recognition. To tackle building recognition, a deep learning architecture is best put to use compared to other approaches such as handcrafted features. Combining low-level features to create high-level representations of the buildings. Determining these features allows for understanding the characteristics of different buildings. A deep learning approach will be applied to recognize certain objects in the historical building photographs of the town of Trnava. The dataset will consist of will personally-taken images by the author.

Image recognition relied on several handcrafted features with a classification system in the past. However, handcrafted features aren't suited for high-dimensional sets of image features. Therefore, Neural networks provide a good solution for automated high-dimension feature extraction. With the computing power available today, it's practical to implement a network with several hidden layers.

Convolution Neural networks use convolution and polling layers with supervised classification.

Standard networks such as LeNet, AlexNet, and GoogLeNet were considered. However, the LeNet model will be chosen due to the limited resource of memory computing. TRNAVA LeNet Model: The TRNAVA LeNet model was first tested. The model was trained with 36 training images and 14 validation images with an image dimension of 28 X 28 with RGB color of type JPG. Through various epoch interactions, the model accuracy resulted in a 50% loss in the validation data. A model with a 50.96% accuracy is not acceptable in today's world. Deep learning architectures should be capable of having an accuracy of 90% or above. TRNAVA LeNet 10 model: After increasing the number of training images, modifications of hyperparameters with the TRNANA LeNet 10 model resulted in accuracy above 80. This model was trained on 460 training images with 140 validation images. The image attributes were kept the same, 28 x 28 pixels with RGB color of type JPG. Though an 80% accuracy may not seem impressive it's important to consider the limited resources available.

A CNN-based approach allowed for automatic high-dimensional feature extraction. Resulting in a great performance with a relatively small dataset. The model was able to detect objects/descriptors in the images of building to recognize historic buildings with high accuracy. The model is extremely scalable. Adding more complex data such as a dataset of photographs of various historical buildings from various angles will refortify the model. Further, the model can be imported onto portable devices with lower computing and memory power while maintaining impressive performance.

Visual data classification in post-event building reconnaissance

This article discusses how a large volume of images from earthquake disasters were used to identify certain structures, the materials they are made out of, and their damages. The program utilizes convolutional neural network algorithms for image classification and object detection in order to identify the object of interest from the inputted images. The article describes data augmentation techniques that were used to develop this program. The parameters in the neural network were trained using a large amount of images in order to achieve robust analysis of the images collected.

CNN's, or convoluted neural network algorithms, made recently have been implemented to extract content of interest automatically from large amounts of perishable visual data. In post-event building reconnaissance teams collect perishable data to gain knowledge after disasters. Potentially generating tremendous amounts of information in short periods of time. For the professionals carrying out this process, it is a tedious and time-consuming task, with only a small amount of data annotated and used for science after analysis. The CNN algorithms reduce the human workload in this process, using image classification and object detection to accurately extract these images in its procedures. Using a large volume of visual data from past disasters, collapse classification and spalling detection occurring on concrete structures included in the visual information that can be detected and extracted.

B.2. Questions

What makes these papers important/relevant?
The papers we have chosen to review all relate specifically to image recognition of buildings. Before beginning the data processing and modeling stages of our project we want to see what researchers in the field have recently found to improve their building recognition results. This way we can possibly implement some of these methods in our project.

How big is the potential market?
The market for this project would be very large. Hypothetically we would start with just implementing the software on the campus of UNCC, but then any college could use the same software for their campus. It could also be used for the downtown areas of large cities, but we would likely need to tweak some things for the system to work on a more urban landscape.

What data is available for testing and/or training algorithms?
For our use case, we want to make a model that predicts UNC Charlotte buildings based on a picture. For this, we will need data consisting of images of UNC Charlotte buildings. Sadly, this data is not available on the scale that we need. Aside from a few pictures of buildings on Google Images, Google Street View, and the UNC Charlotte website, there is not nearly enough data available online. For this reason, we will need to gather data ourselves by taking and compiling pictures of buildings and labeling them.

What companies are solving similar problems to yours?
While there are no companies working on building recognition for the UNC Charlotte campus, there are some working on building recognition for other uses. The Royal Academy of Arts in London launched an app in 2017 called “Smartify.” Smartify is an app that allows users to take pictures of art pieces and buildings and receive information about scanned objects. While the app more focuses on painting recognition, building recognition is still utilized. This is just one example of a company using building recognition in their products. On top of this, there is also research still being done on building recognition models.

What open source code is available for our topic?
There are not many codebases specifically designed for building detection, however there are several libraries that can be used in order to facilitate building detection. The main library and most popular is OpenCV, which is widely used computer vision and machine learning library. While it features many capabilities such as facial recognition or motion tracking, it also features object detection methods which can be used in order to create models for our product. Keras is a API created by Google that provides an interface for artificial neural networks. The key selling point being that Keras is designed for user friendly network implementation.

How active are the communities surrounding the code?
OpenCV is a widely used library that is constantly being updated and monitored. The GitHub alone has over a thousand contributors and tens of thousands of commits. It also has a dense FAQ section and frequently updated documentation. Meanwhile, Keras being created by a Google engineer means the library already has a strong foundation. Their website is home to a thorough code example section as well as well written documentation. The project also has a community of over a thousand contributors on GitHub with improvements being made daily.

What problem will your Computer Vision solution solve, and for whom?
The goal of our product is to help recognize buildings across the UNCC campus. Our product will be a model that takes in images of buildings and determines the name of the building. Our product will be most useful for the various UNCC departments. For example, the UNCC touring department can implement the model into their tour guide which makes exploring campus more interactive.

What are their results and how did they achieve these results? (Academic Research Paper)
The model described in the ”Building recognition system based on deep learning” article used a neural network approach that had an accuracy above 80% in recognizing historical building in the town Trvana with a limited dataset and memory computing. The model takes a CNN-based approach with mutiple layers (Convolution layer, sub-sampling layer, convolution layer, sub-sampling layer, fully

connected MLP). The author suggests acquiring complex data of the buildings such as different angle view, distance, and lighting is extremely crucial to success of the model. In addition, the author used the LeNet 10 architecture due to its efficient memory computing. Therefore, allowing this approach to be practical for portable devices with lower memory computing capabilities without sacrificing model performance.

What value will it provide them? What are their pain points?
The software will detect and identify buildings on the UNCC campus using an image of the building. This will allow users to receive important information about buildings around campus simply by taking a photo of the building that they can take with their phone. This product will benefit the staff, students, and guests at UNC Charlotte. This service is especially vital to new students and visitors who are not familiar with the campus. Some potential pain points for customers could be the price of the product, or the ease of use of the application. In order to mitigate these issues, the product could be made easily accessible, and the usability of the application should be a priority.

How are potential customers dealing with these issues now?
Currently, if those visiting the UNCC campus want information about a certain building, they will first have to identify the building by finding a sign indicating which building it is. Then, they would have to use their device to look up the name of the building and find information about it on the UNC Charlotte website. With our product, customers will only have to input an image of the building they want to identify by taking a picture with their phone, and the application will return the name of the building along with any other information customers may want to know.

C. Data Collection

During this stage of the project, we collected and compiled our dataset, and then processed and labeled the data to prepare the dataset for the modeling stage.

C.1. Collection Method

A dataset containing images of buildings across the UNCC campus does not exist, so our team created our own dataset of images using pictures taken by our own devices. Twelve popular buildings on campus were selected to create the dataset with and several group members went around campus and took pictures of the buildings in both day and night time lighting conditions. Pictures of the buildings were also taken from many different angles in order to create a more robust training set. In total, our group collected over 2,700 images for the twelve buildings. Each image was resized so the entire dataset consists of images with the dimensions of 1920x1080 pixels.

C.2. Data Visualization

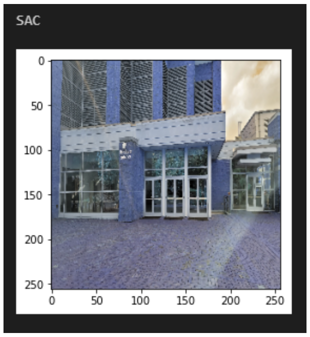


Figure 1. An example of a photo in our dataset

C.3. Data Annotation

To assign the correct building it's proper label, we imported all the images into our Jupyter workspace and created a dictionary based on the names of each of the buildings within the dataset.

D. Modeling

During this stage of the project, we referred to our literature review for ideas, performed experiments, trained our model, and assessed the model's accuracy.

D.1. Description of Model

The model uses a tuned pre-trained ResNet50 Architecture with the last fully connected layer trained with our dataset containing pictures of UNCC campus buildings. In order to measure the performance of our model we used a cross-entropy loss for multiclass image classification. While all loss functions penalize accuracy measurements for incorrect predictions, when using Cross-Entropy, the accuracy is penalized greatly when confidence is high but the prediction is incorrect. Thus, we believed this to be the best way to measure our model's loss. With regard to Hyperparamters, they varied during experimentation but for the final model we utilized a learning rate of 0.0001, batch size of 32, and a number of epochs equal to 15. The use of this model led to an accuracy of 86% when tested.

D.2. Experimentation

From our literature review we found that, according to Bezak (Bezak. 2016), using a pre-trained neural network would most likely produce the best results for our objective of building identification. In our experimentation, several different networks were tested, as well as a CNN developed from scratch. Originally, the use of GoogleNet's pre-trained model was used in order to achieve an accuracy of just over 75%. We saw this as an easy starting point due to the ease at which GoogleNet can be implemented. We attempted to use other networks to see if a higher accuracy could be achieved, so next, we implemented the Visual Geometry Group (VGG) network. In particular, we used the architecture that contained sixteen convolutional layers as opposed to the layer that contained nineteen. Again, this method was simple to implement, however this only produced an accuracy of about 70%. Then we attempted to implement a self attention and vision transformation process on top of a ResNet50 foundation, however this produced an accuracy of 50%. Not satisfied with the accuracy achieved so far, we attempted to create and train our own CNN. Although our research indicated that a pre-trained model would be more effective, a handmade network was created. Our research was confirmed when our network was only able to achieve a 50% accuracy. Finally, our last experiment was similar to our final model, but unlike our final model, this experiment was conducted without the introduction of any data augmentation. In our final experiment, we were able to achieve an accuracy of over 88%.

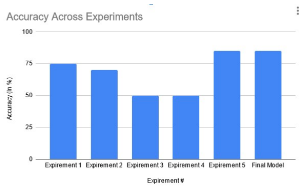


Figure 2. Accuracy from experimental tests

D.3. Literature Review

The motivation behind our approach of using a pre-trained neural network for modeling image classification for the UNCC building comes from an article regarding building recognition, ”Building recognition system based on deep learning” by Pavol Bezak. The author has similar conditions for collecting building information and modeling it to predict buildings. For the data collection, our group had to hand-collect pictures of the respective UNCC buildings identical to the author who takes images of historic buildings in the town of Trnava.

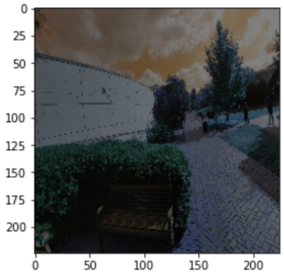
The article discusses how real-life conditions make building recognition often a complex problem. The varying lighting, angle, and resolution are often major factors contributing to the complexity of building recognition. To tackle building recognition, a deep learning architecture is best put to use compared to other approaches such as handcrafted features. Combining low-level features to create high-level representations of the buildings. Determining these features allows for understanding the characteristics of different buildings. Compared to handcrafted features that aren't suited for high-dimensional sets of image features.

To implement a neural network to classify the buildings, a CNN-based approach allowed for automatic high-dimensional feature extraction. Resulting in a great performance with a relatively small dataset. The model was able to detect the images of building to recognize historic buildings with high accuracy.

The goal of this project lines up well with the author's goal of building classification with a relatively small dataset. Hence, we steered towards a CNN-based classification approach for our project rather than using handcrafted features. We implemented a pre-trained CNN approach for our modeling our data. However, compared to the author we aren't as bottlenecked by computing power. Therefore, we increased the image size and used various other heavy-weight pre-trained networks compared to the author's 28x28 image size that was fed into a lightweight LeNet.

D.4. Results

As mentioned previously, our final model was able to achieve an accuracy of 86% when evaluated. Given an input image, the model will output a predicted label for the building.



```
Prediction:  34 - Prospector
```

Figure 3. Example out put when an image of Prospector was given as input

E. Conclusion

Building detection is a situation within computer vision that comes with several unique challenges. Creating a dataset from scratch is beneficial, in that, while we were collecting the data, we had to keep in mind and address several of the scenarios in which the building detection may be more difficult, whether it be difference in lighting, angles used, or even weather. Overall, we are satisfied with the complexity of our dataset as well as the accuracy produced by our final model.

F. Contribution per Team Member

Madison Melton: Team Formation, Research, Dataset Collection, Reporting, Presentation

Rachel Miranda: Proposal, Research

John Taylor: Project Idea, Research, Dataset Collection, Reporting, Presentation

Prasheeth Venkat Kumar: Research, Dataset Collection, Annotation, Modeling, Reporting

Nathan Williams: Research, Dataset Collection, Annotation, Modeling, Reporting

G. References

Bezak, P. (2016). Building recognition system based on deep learning.*2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*

Guo, X., Li, Y., & Ling, H. (2016). LIME: Low-Light Image Enhancement via Illumination Map Estimation.*IEEE Transactions on Image Processing*

Li, J., & Allinson, N. (2013). Building recognition using local oriented features. *IEEE Transactions on Industrial Informatics.*

Yeum, C. M. (2018). Visual data classification in post-event building reconnaissance.*Engineering Structures*

Zhang, C., Yankov, D., Wu, C., Shapiro, S., Hong, J., & Wu, W. (2020). What is that building?: An End-to-end System for Building Recognition. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*