**Project title**



*Federated* *Learning on Morphology Data*

Noteamname

Mason Jerome

Mayuresh Hooli

Date:

# Executive summary (1 paragraph; 5 points)

The

The executive summary is a brief description of the project, and should include a quick overview of (1) project background, (2) need, goal and objectives, (3) design and implementation, and (4) expected results and benefits of the project. Since the executive summary is a summary, it should be written last.

# Introduction

## Problem background

Town planning has been one of the most important aspects of human civilization. The concept of town planning has existed since at least 8000 BC. Ancient civilizations like the Indus Valley Civilization had civic zones since the 3rd Millennium BC. As the human race has evolved, so too have the cities. The small self-contained towns in the ancient era have given way to sprawling megacities in the current world. The cities in the US were built around the zoning laws in the 1940s with companies such as General Motors having a major say in the urban planning. As the plans for these cities are getting outdated in the 21st century, urban planners are looking at new ways to develop these cities.

## Needs statement

A lot of cities that have been built, were based on the incorrect ‘car first’ principle. The United Nations Sustainability Development Goals for 2030 talks about sustainable cities in their 11th point. According to the report, the global urban population surpassed the rural population for the first time in human history in the year 2008. By 2050, two-thirds of the world population will be living in urban areas. As the cities are growing at a rate of 73 million new urban residents every year, urban planners are finding it difficult to survey areas of this ever-increasing sprawl. There is a need to identify a new way to classify urban morphology. This can be done using Computer Vision.

## Goal and objectives

The goal of this project is to develop a machine learning model to classify different classes of urban sprawl.

The conventional way to classify an urban sprawl is by manually surveying the location and making an inference. To an untrained urban planner, the inference may or may not always be right. The other issue with this is that surveying an area manually takes a long time as the person may have to travel to different places to get the ideal reading. The solution of using federated learning for urban morphology removes the need for an urban planner to travel to different places to identify the type of morphology. As the algorithm has already been trained, the issue with an untrained eye trying to check whether the location falls under a certain class does not exist.

The objective of this project is to create a model for urban morphology which will give us the right class with a certain level of accuracy. As the model requires a large data set, which we do not have at the moment, the threshold we have is 75% for each of the 10 classes.

## Design constraints and feasibility

Creating a model for urban morphology requires a large data set. Based on an initial estimate and discussions with domain experts, we would need an estimated 1500 images for each class for the training set. The data we obtained from Microsoft Bing maps is free only up to a limited number of images with a corporate account, while none of the images are free using a student account. Google has better data, but it is not available for free and the terms and conditions state that using their images gives them the right to own this IP. Thus, the data set available for this project is small. Nonetheless, we can develop a model for these images.

# Literature and technical survey

The cities today are sprawling far and wide, making the conventional surveying techniques difficult. Thus, many new techniques are being proposed for identifying the morphology. Based on our research, we have found 3 techniques that are being used for this:

* Classification using drones
* Machine Learning
* Ensemble Machine Learning

The use of drones for urban planning has been gaining traction since the early 2010s. This has saved a lot of time and money for planners who would travel to various locations to identify the class of urban area. Various companies, such as diji, have created drones that will survey and photograph part of a map and give the data. The other major breakthrough over the last few years has been the mapping software, particularly Google Maps. Google maps provides a comprehensive view of any location, as well as street view. Obtaining this data would help planners plan for any location. Unfortunately, these innovations have not translated to improving performance for untrained planners. The planners still need to go through all the images and classify each of them individually, which, despite reducing the time required, is still not as efficient.

Companies such as T-Mobile, which deploy their network radios depending on the urban planning and morphology decided to experiment with Deep Learning Algorithms. They obtained 1500 images on each of the 10 classes and deployed a model to identify the classes. This model used object detection algorithms built on top of resnet to identify the classes. With the large amount of data available at their disposal, the accuracy was high as well.

Other companies like AT&T, went for a similar approach, but with a smaller data set. As the large amounts of data were not available, the initial model gave a poor accuracy, so 3 different models were developed, one for urbanness, the second for foliage and the third for types of trees. Each of these models used retinanet built on top of resnet to give a classification model. Once these models were developed, another classification component was added to this, to identify the likelihood of the image belonging to a particular class. Based on this, a voting algorithm was used to classify the images into various classes.

# Proposed work

## Evaluation of alternative solutions

We received a working code of the alternate solutions. The two techniques, Machine Learning and Ensemble Machine Learning, were quite effective, but not without its faults.

Machine Learning was the most logical approach to solve this problem. As we move towards the future, most of the problems that we are solving will involve machine learning in some way or another. Thus, for urban planning, the use of a computer vision seemed to be a logical fit. Following are the pros and cons of this solution:

* Pros:
  + Machine Learning is a logical fit and will improve the accuracy of the prediction
  + It will save a lot of time, thus helping urban planners make quick decisions
* Cons:
  + Computer vision algorithms depend on the quality and quantity of data. A large dataset may not always be available
  + A general machine learning model may not be able to look at the problem from all angles

Another solution that we looked at was Ensemble Machine Learning. This technique has been used by AT&T to identify the right places to set up new basebands. This model was significantly more accurate than the machine learning model. Following are the pros and cons:

* Pros:
  + Higher accuracy with a lower data set size
* Cons:
  + Model is extremely complex

The decision to go with Federated learning stems from the fact that machine learning models are already being used to solve this problem. The Machine Learning model using resnet is good, but requires a lot of data, while the model developed using ensemble learning is extremely complex to retrain on different urban settings. Federated learning helps us look at every angle of an urban area and classification can be performed on each of those angles.

## Design specifications (2 pages, 15 points)

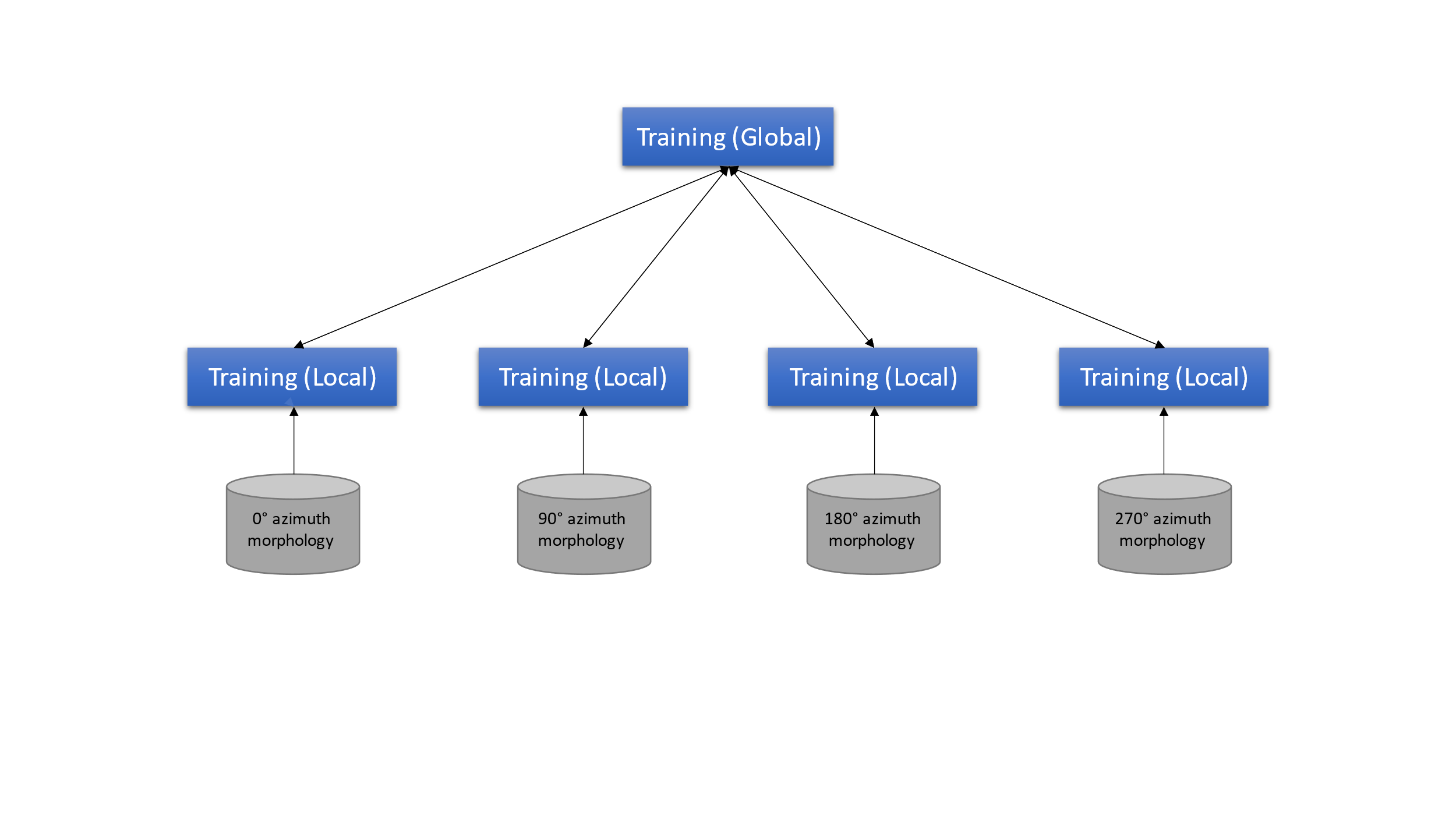


Figure 1. Federated Learning flow diagram for urban morphology.

Once you have identified a solution that addresses the needs of the project, it is time to present the specifics of your design. Start with a high-level block diagram of the system (i.e., 5 building blocks or modules), followed by a description of each module. This description should include techniques (e.g., algorithms, devices), parts (e.g., hardware, software), and the “glue logic” that will make the *system* work. Your proposed design should build support for the *feasibility* of your project.

Below are some examples of good and bad block diagrams.

|  |  |
| --- | --- |
| (a) |  |
| (b) |  |
| (c) |  |

Figure 1. (a) A really bad block diagram. First, the text is practically unreadable.   
Second, the diagram is so generic it could represent anything. (b) Another bad block diagram. Why? (c) A good block diagram. Do you get an idea of what the system depicted is about, and what its main components are?

## Approach for design validation

To ensure that the trained Federated Learning model was working, it needed to be validated on an independent image set. For this, we split the image dataset into the training set, which went to each respective clients’ dataset depending upon the angle at which the image was taken, as well as a testing set which comprised 20% of the images and was not used for any training purposes.

After each round of client training and server Federated averaging of the client models, the global model was evaluated on the test set to provide a validation accuracy on the images.

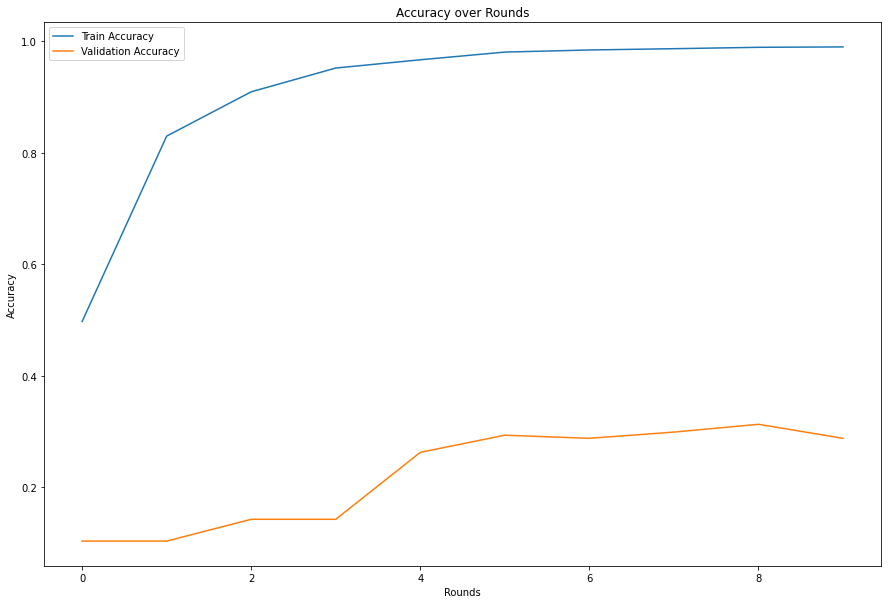


Figure 2. Design validation for the project.

The validation accuracy starts off at just over 10% which makes sense as there are exactly 10 classes which the model can predict. Over the iteration of training rounds, the model converges at between 30-35% classification accuracy on the validation set of data.

While this accuracy is not as high as we would have liked, the model is limited by the dataset size. We were unable to collect as many images for each class as necessary to decrease the overfitting of the model to achieve a greater generalization. The images for each class that did collect ranged from 133 to 249. Ideally, we would have 1000s of each class to train on.

We utilized the same relevant hyperparameters used for the Federated training on a standard Convolutional Neural Network with the exact same architecture used for the client training. This model achieved similar validation accuracy, showing that the Federated training was working properly.

For the demo, showcasing the validation results will have to work as the model takes too long to train for it to be fit into the demo’s timeframe.

# Engineering standards

## Project management

The team consists of 2 members:

* Mason Jerome
* Mayuresh Hooli

Following is the task list we followed for our project:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Task Description | Mayur | Mason |
| 1 | Literature Survey | ✓ |  |
| 2 | Problem articulation | ✓ |  |
| 3 | Data Collection | ✓ | ✓ |
| 4 | Expert Analysis | ✓ |  |
| 5 | Solution Flow Design | ✓ | ✓ |
| 6 | Data Set Creation |  | ✓ |
| 7 | Model Building |  | ✓ |
| 8 | Model Testing |  | ✓ |
| 9 | Comparison of Results |  | ✓ |
| 10 | Documentation of Results | ✓ | ✓ |
| 11 | Project Report | ✓ | ✓ |

## Weekly schedule of tasks, Pert and Gantt charts

Figure 3. A Gantt chart for the project.

## Economic analysis

This project is very similar to various industry projects; thus, the economic analysis of this project has been performed. Following are the things we have considered:

* Economic viability:
  + Understanding the morphology of an urban neighborhood is a highly desirable topic, not just for urban planners, but also for telecom companies, construction companies and various other industries.
  + Recently, a similar project was undertaken by AT&T that was valued at millions of dollars. The true amount is classified information.
  + Since this is a software development project, the cost was 2 engineers working full-time on this project for 40 hours a week.
  + At 4 weeks, the cost is 320-man hours.
* Sustainability:
  + The product has been developed using open-source software.
  + The code has been designed to be modular.
  + There are no major challenges in maintaining the software.
* Manufacturability:
  + Being a pure software project, there are no manufacturing challenges
  + The software is being designed for scaling

## Itemized budget

* Hardware Cost:
  + Server: Free Google Colab Server
* Software Cost:
  + Microsoft Bing Maps: Free (for us)
* Development Effort:
  + 100-man hours

# References

1. Lilley, K.D., 2009. Urban morphology.
2. Pan, W. and Du, J., 2020. Towards sustainable urban redevelopment: urban design informed by morphological patterns and ecologies of informal settlements. In Urban Ecology (pp. 377-411). Elsevier.
3. “Sustainable Cities and human settlements | department of economic and social affairs,” United Nations. [Online]. Available: https://sdgs.un.org/topics/sustainable-cities-and-human-settlements. [Accessed: 10-Dec-2021].
4. Najafizadeh, L. and Froehlich, J.E., 2018, October. A feasibility study of using Google street view and computer vision to track the evolution of urban accessibility. In Proceedings of the 20th international ACM SIGACCESS conference on computers and accessibility (pp. 340-342).
5. Dietterich, T.G., 2002. Ensemble learning. The handbook of brain theory and neural networks, 2(1), pp.110-125.
6. Li, T., Sahu, A.K., Talwalkar, A. and Smith, V., 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine, 37(3), pp.50-60.

# Appendices

## Bio-sketch (one paragraph)

The members of the team have been working in Machine Learning and IoT for some time.In addition to this, one of the team members has worked on a similar project in the telecom domain.