Emergency Planning for Aerial Vehicles by Approximating Risk with Aerial Imagery and Geographic Data

Caleb Harris*, Seulki Kim†, Alexia P. Payan‡ and Dimitri N. Mavris§
Aerospace Systems Design Laboratory, School of Aerospace Engineering,
Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA

Urban Air Mobility and Advanced Air Mobility require the certification of novel electrified, vertical takeoff and landing, and autonomous aerial vehicles. These vehicles will operate at lower altitudes, in more dense environments, and with limited recovery abilities. Therefore, emergency landing scenarios must be considered broadly to understand the risks in some situations of flight failures. This work provides a preflight planning tool to assist these vehicles when emergency landing is required in crowded environments by fusing geographic data about the population, geometric data from lidar scans, and semantic data about land cover from aerial imagery. The Pix2Pix Conditional GAN is trained on Washington D.C. datasets to predict eight classifications at a 1m resolution. The information from this detection phase is transformed into a costmap, or riskmap, to use in planning the path to the safest landing locations. Multiple combinations of the cost layers are investigated in three test scenarios. The Rapidly Exploring Random Tree (RRT) algorithm efficiently searches for an alternative path that minimizes risk during emergency landing. The tool is demonstrated through three scenarios in the D.C. area. The objective is that the tool allows for the safe operation of UAM and AAM vehicles through crowded regions by providing confidence to the local population and federal regulators.

I. Nomenclature

\[
\begin{align*}
UAS &= \text{unmanned aerial system} \\
UAM &= \text{urban air mobility} \\
AAM &= \text{advanced air mobility} \\
evTOL &= \text{electric vertical takeoff and landing aerial vehicle} \\
EFB &= \text{electronic flight bag} \\
CNN &= \text{convolutional neural network} \\
GAN &= \text{generative adversarial network} \\
RRT &= \text{rapidly exploring random tree}
\end{align*}
\]

II. Introduction

The use of aerial vehicles for transportation and mobility in urban areas has motivated a change in the design of traditional rotorcraft. Conventional rotorcraft in urban areas have been plagued by noise, pollution, and high operating expenses. However, the latest technological developments in battery energy storage, electric propulsion, and automation of flight vehicles have enabled traditional designs to evolve. As a result, many aerospace industries have adopted and invested in new concepts of vertical lift vehicles for urban air mobility (UAM). These vehicles are designed for passenger transportation, cargo logistics, and emergency response with automated flights around metropolitan areas.

This rapid growth in novel vehicle designs raises the concern of the safety of flight over urban environments. In particular, deciding where and when to land requires a high level of understanding of the environment to avoid collisions with property and people in the area. Traditionally, human pilots have used preflight and inflight tools and techniques,
as detailed by the Federal Aviation Administration (FAA) [1] and the European Union Aviation Safety Agency (EASA) [2], to find safe areas for landing when an airport or a heliport is not available, or when an emergency occurs. There have been recommendations on how to perform reconnaissance of an area in preparation for unknown situations and on how to get visual information on an area using tools such as Google Earth. In the past decade, the advancement of electronic flight bag (EFB) s and flight decks in the cockpit have added the potential for advanced visualizations and technology to provide pilots with safe emergency landing locations. This has started to shift in the last couple of years with autonomous methods to perceive, detect, fly, and land these vehicles, as demonstrated by Reliable Robotics and Daedalean [3]. There is also an academic interest in automated methods of finding landing sites [4] and of responding to in-flight emergencies [5].

Previous work in Harris et al. [6] formed a database of obstructions in the environment, with the initial focus on medium and low-voltage power lines. A hybrid approach was necessary to combine the visual features of utility poles using the U-Net architecture and the spatial relation of the electrical grid network, using a many-to-many path predicting. Additionally, previous work in Harris et al. [7] sought to exploit machine learning techniques to discover suitable landing zones from aerial imagery and lidar intensity maps. A convolutional neural network was trained to predict the land cover type. A K-means++ clustering algorithm was then used to classify three square-inch regions of the map into hazardous zones, high-risk zones, or safe landing zones. The single-class prediction technique, the U-Net architecture, and the limited amount of data are items in which improvements can be made to produce more accurate and informative results.

The framework is to be used as a preflight planning tool to assist these vehicles when emergency landing is required in crowded environments. This is done by fusing geographic data from the population, geometric data from lidar scans, and land cover predictions from aerial imagery. The worst-case and best-case scenarios can be prepared by using visual and statistical data on the obstacles, terrain data, land type, population risk, and flight path risks throughout the environment. The path planning module provides the safest and quickest trajectories using the predicted costmap, but this data is supplemented visually with terrain and population risk data. Throughout the tool, a series of checks and balances in a decision tree provides logical decisions, which can be replaced with a human operator to decide where the best landing sites are located and how to navigate to these spots. The key requirement in this framework is to approximate the costmap, or riskmap, of the environment. This is because the costmap will decide where and how to land the vehicle during the path planning stage.

The main contribution of this work is the development of a preflight planning framework that allows users to investigate and evaluate alternative routes in preparation for in-flight emergencies. The costmap formation and path planning tasks are examined with different algorithms and three scenarios. The computational cost of the algorithms is discussed to provide a baseline for future use for preflight planning.

III. Background

A. Aerial Vehicle Emergency Planning

A majority of work in the past decade has focused on using available population data and elevation maps to form costmaps, or riskmaps, to plan a safe and optimal path, such as work by Primatesta et al. [8]. Similarly, Carney et al. [9]
developed a methodology and framework for combining population data and elevation data to find safe landing strips for UAVs. However, in recent years, the use of visual information has increased because of the availability of data and the capability of computational resources. For example, Ayhan et al. [10] took advantage of Google Map images by using edge detection algorithms to find candidate landing sites and then combined this with elevation and land type data to determine feasible landing spots. Semantic segmentation methods in particular have taken the lead in remote sensing areas because of the high-resolution outputs. This is demonstrated in the work by Montoya-Zegarra et al. [11], where semantic segmentation is used to predict from a set of urban land type classifications. These classifications were used for road networks by combining with other datasets in an additional step that the authors call “inference”.

The past works have thus far focused on predicting urban land cover or the formation of obstacle maps; however, there is the potential for utilizing the features within the network to approximate flight risk. For this to be demonstrated, there is a need for additional background into the best techniques for semantic segmentation to provide these informative features and into data fusion methods to integrate these features with other geographic features. This work focuses on applying this to aerial system emergency landing and planning.

B. Computer Vision and Deep Learning

The field of computer vision has changed rapidly with new ways of extracting features from visual information. The growth of accessible satellite and aerial imagery has provided the data to train convolutional neural networks to find informative decision-making and object detection features. However, consideration of computational time and data storage limits the design of these architectures and the size and format of the datasets. Research into network architectures continues to advance and produce new models and training techniques. For example, U-Net is a CNN architecture that features an encoder and decoder section and can produce pixel-wise classifications for semantic segmentation [12]. This was used for previous work in this area and is a powerful technique to produce high-resolution classifications of the environment. However, the training and implementation process can be improved by new and improved techniques. One such method is generative adversarial networks, or GANs, such as Pix2Pix [13]. This model is used in this work because of its proven capability to map three channel images to higher-dimensional feature sets, as seen in the Pix2Pix paper with images to OpenStreetMap features. Furthermore, the use of a GAN may show performance improvements for a limited and noisy dataset.

C. Sensor Fusion of Geospatial Information

The wider availability of data promotes a data-centric view, where more time and focus is spent on forming the datasets for training, as explained by Andrew Ng in [14]. The massive increase in geographic data, such as population density, traffic movement, and infrastructure maps, provides additional insight into the predicted state of the environment and how suitable it may be for a safe landing. The difficulty lies in how this information can be used together rather than separately. Ten Harmsel et al. [15] used public databases of New York City, which included population density, structure elevation, and terrain data. The population distribution was collected from the U.S. Census and transformed into the population density map. The structure and terrain datasets were combined to incorporate building height and terrain metrics into a riskmap for emergency landing planning. Compared to other papers devoted to dealing with static demographic data, Di Donato and Atkins [16] employed a mobile phone database to monitor moving people and predict the area occupancy in real-time. The collected mobile phone data combined with census data provided an accurate risk assessment for emergency landing planning. In this paper, geographic data of population and structure heights over Washington D.C. is used to form a costmap or riskmap.

IV. Methodology

The planning tool makes use of available geographic and visual data for an aerial emergency planning. The data includes structure heights, demographic data, terrain slope, and urban region landcover labels. The combination of these data layers, excluding structure heights, forms a riskmap, approximating a surface function of the likelihood of failure to safely land, therefore steering any path away from maximizing the risk if contingency planning is required. The structure heights remain as obstacles to be avoided in path planning. For data acquisition, the building heights and demographic data are collected from the OpenStreetMap and Washington D.C. open datasets, respectively. The urban region labels are predicted using visual data from satellite or aerial flights. A deep-learning approach is used for semantic classification using the Pix2Pix network. A risk-aware path planning algorithm is then used to explore one or more safe paths that avoid obstacles to the target location at a predefined altitude. Several paths are planned
based on multiple costmap creations to see how risks are fluctuated with variations in weightings over the map and how they impact path planning. Three scenarios are used to test this hypothesis and perform subjective validation of the risk-aware path.

A. Process and Tools

The Washington D.C. Open Data[7] and D.C. From Above aerial imagery[8] datasets provide the geographic and visual data to perform training and testing for the costmap creation. The Google cloud infrastructure is selected for the framework because of the available data storage, GPU computation, and geospatial data analysis capabilities. Google Drive, Colab, TensorFlow, and Earth Engine combine into a robust framework of tools. Google Drive allows for data processing in Earth Engine and Colab that is entirely in the Google Cloud. TensorFlow is a library developed for conveniently and efficiently designing and training neural networks. Colab accesses Google GPUs and TPUs in the cloud infrastructure through a Jupyter-notebook, python-code environment. Lastly, Earth Engine provides computation and visualization tools for geospatial data analysis and machine learning. Google Earth Engine is documented in detail by Gorelick et al. [17].

B. Creating Data Layers

1. Population Density

Population was acquired from the Washington D.C. Open Data, which freely shares hundreds of government datasets with the public in the District of Columbia. The 2020 Census Blocks is the smallest polygon blocks for census and this data was collected for the latest and detailed geographic census in this region. Each block includes several metrics, such as total population, population of specific races and ages, land area and length, and more. The population density was then calculated by dividing the total population by area in each polygon. Additionally, the coordinate reference system of the data was converted from WGS 84 (EPSG:4326) to UTM zone 18N (EPSG:32618) so that the unit of measurements is changed from a degree of latitude and longitude to a meter. Finally, the population density calculated over the polygons was rasterized with 1-meter by 1-meter pixels for the riskmap generation, shown in Figure 2.

![Population density expressed in vector polygons](image1)

(a) Population density expressed in vector polygons

![Population density rasterized with a 1-meter resolution](image2)

(b) Population density rasterized with a 1-meter resolution

**Fig. 2** Rasterization of population density over the entire Washington D.C.

---


2. **Obstacles**

The primary focus of this work has been in forming a costmap or riskmap; however, in order to provide the user with a higher level of confidence in the flight plan, this work also included an obstacle map that aerial vehicles cannot go through and should avoid. While it is assumed that most flights will be above the obstacles for a majority of the flight, the assumption is that a vertical landing may be required at any moment, thus requiring a vertical clearance between obstacles. Therefore, the obstacle map must be formed for structures and obstacles above a certain height.

There are multiple ways to account for these obstacles in the environment. One is to use any data available on buildings and utility poles in the area. This work utilized open-source geographical data from OpenStreetMap (OSM) to collect available building data. The building data over the Washington D.C. was collected through Overpass Turbo API that helps extract OSM data easily and quickly, as shown in Figure 3. The collected data provides building heights and levels (floors) and some structures have either heights or levels. For buildings possessing only level data, their heights were calculated by their levels multiplied by a height per floor, which was assumed as 4 meters. The building heights data was also converted from WGS 84 to UTM zone 18N and rasterized with a 1-meter resolution.

![Fig. 3 Building data extraction using open source OpenStreetMap](image)

The potential problem in collecting available structural data through OSM is that there is not enough data, which are limited for specific regions. An additional way to augment the data deficiency is to use data on the terrain elevation since this is easier to collect consistently. However, LIDAR data is often used for high-resolution terrain maps that can be computationally expensive to collect and process. However, Digital Surface Models (DSM) can be collected from LIDAR or other sources and used at the processed resolution. For this work, the Digital Surface Model mosaic from the Washington D.C. Open Data was used. This DSM is at a 1-meter resolution and was created from a LiDAR dataset during a flight on April 5, 2018.

3. **Urban Semantic Segmentation**

The costmap formation process requires an accurate prediction of the type of area. When an aerial vehicle operates above a region, whether actively landing or preparing for a future emergency event, it is essential to know what terrain is below the vehicle. A deep learning-based approach to semantic classification is powerful as it allows for continuous improvements of labels with updated datasets, improves the prediction resolution from human-generated labels, and allows for ‘on the fly’ calculations rather than storing offline maps. Therefore, this approach was selected to produce urban landcover predictions.

The data labels for a region in Washington D.C. can be seen in Figure 4. The current dataset includes the labels for structures, greenspaces, roads, parking lots, sidewalks, water, trees, and poles. The dataset was formed from publicly available datasets from the Washington D.C. Open Data. The data was rasterized to label images and used for training, as seen in Figure 5.

The Pix2Pix conditional GAN model was used to perform semantic segmentation on the aerial imagery. The Pix2Pix process, shown in Figure 5, requires a generator, or and encoder-decoder network, and a discriminator network. The Pix2Pix model in this work made use of a modified U-Net for the generator and the original PatchGAN classifier detailed in the Pix2Pix paper.

Two data labeling schemes were used during training and testing with the Pix2Pix framework to determine the best techniques for this dataset and applications. First, an 8-channel prediction was used to predict each classification likelihood for each of the 8 class types. Then, an RGB prediction color encoding was defined to label each of the eight
Fig. 4  Urban landcover labels for Washington D.C. area

Fig. 5  Pix2Pix model
Table 1  Evaluation metrics for semantic segmentation Predictions

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>Evaluation Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel mean-square error</td>
<td>$(1/n)(1/m) \sum_{i=1}^{n} \sum_{j=1}^{m} (Y(i, j) - \hat{Y}(i, j))^2$</td>
</tr>
<tr>
<td>Pixel Accuracy</td>
<td>$TP_i + TN_i/(TP_i + TN_i + FP_i + FN_i)$</td>
</tr>
<tr>
<td>F-score</td>
<td>$2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$</td>
</tr>
<tr>
<td>Mean Intersection-Over-Union</td>
<td>$TP_i/(TP_i + FP_i + FN_i)$</td>
</tr>
</tbody>
</table>

classifications to an RGB code. For example, sidewalks can be defined with the ‘gray’ color encoding represented by: Red: 128, Blue: 128, Green: 128. Figure\[2\] provides a concise representation of the labeling. The 3-channel prediction is more stable, converges quicker, and is more easily verified during training.

The training process involves 256x256 pixel patches at a 1-meter resolution, meaning each patch has approximately a 256 m² area. The created dataset totals over 44 km² with the training, validation and testing sets split to 30.5, 6.5, and 4.8 km² respectively. The split results in a total of ten thousand training and two thousand evaluation samples. A batch size of 1 is used, which slows the training process but was recommended by the Pix2Pix authors. The training data is sent to Google Colab as a 6-band image, three feature and three label bands, or an 11-band image, three feature and eight label bands.

Qualitative evaluation can be done very easily using the 3-channel RGB label predictions. This is one of the benefits of the method and allows for quick estimates of what the network can and cannot do, as long as the labels are human-interpretable. Four quantitative evaluation metrics are used, seen in Table\[1\]. The first quantitative matrix is the pixel mean-square error, which is calculated as seen in Table\[1\]. This metric is not often used for semantic segmentation methods since the goal is to learn a generative ability, and the mean squared error leads to overfitting. T.P., or true positive, is the successful classification of the correct label, whereas T.N., or true negative, is the correctly not classifying a wrong label. Similarly, F.P., or false positive, is the failure to classify the correct label, while F.N., or false negative, is the failure not to classify the incorrect label. Recall, often referred to as the sensitivity is $TP/(TP + FN)$ and precision, or accuracy, is $TP/(TP + FP)$.

After training each method for ten hours on the Google Colab Pro server using an Nvidia Tesla T4 GPU with 16GB RAM, the 3-channel RGB encoding was selected as the best approach. This is because of improved convergence of the network and more success on the less frequent classes in the unbalanced datasets.

C. Costmap Creation from Layers

Each layer detailed in the previous section is combined into a single costmap using a data fusion process. This requires aligning all layers in the same reference frame, normalizing data into the same scale, and leveraging any information gained from specific layers, such as how traffic data relates to the landcover classification of roads. This combination process, such as in Figure\[6\], results in the costmap, or riskmap, layer with scalar float values in each cell up to the resolution of the datasets.

Once the images have been aligned, filtered, and normalized, the final step is to combine the layers with what can be called a risk function. Previous works such as \[18\] linearly combine the layers, therefore assuming a linear risk function. The idea here is that the real risk in the environment is being approximated. It approximates because the data is noisy, and the risk is a highly nonlinear representation of many factors.

D. Path Planning

The availability of an accurate costmap in discrete space allows for a wide range of algorithms to be used to find the optimal path. The Rapidly Exploring Random Tree algorithm, or RRT, produces trajectories that minimize risk to one or more of the landing zones. The RRT algorithm can be implemented in various ways by modifying the sampling method or forward propagation model. In addition, powerful techniques have been used to provide alternative routing for landing sites \[19\]. Therefore, the RRT algorithm is explored for the best parameters and implementation when applied to this framework. The energy-aware RRT from the algorithm from Lee et al. \[20\] provides an efficient search algorithm that can handle a dynamic riskmap and uses an energy-based tree expansion.
V. Results

The following section details the results for the semantic segmentation of urban landcover, costmap formation, and path planning steps. The Pix2pix GAN produces label predictions for the training and prediction datasets, and the pixel accuracy and Intersection-over-Union provide insight on the success. The costmap formation process creates 20 costmap approximations that can be used for sensitivity analysis or alternative routing. Then, the vanilla A* and RRT algorithms are compared to the energy-aware RRT algorithm in speed and performance when planning in the created costmaps.

A. Semantic Segmentation

The Pix2pix GAN produces the urban landcover predictions from semantic segmentation of aerial imagery. Examples of the outputs after a limited training session are shown in Figure 7(a) and outputs after the full training session are shown in Figure 7(b). After the entire training of 500,000 iterations with 2000 samples, demonstrate the ability of the network to learn the labels shown here. The evaluation metrics provide further insight that the network is learning the correct features to predict the label from the images. The labels which still show trouble in learning are water, trees, and parks. The primary reason for poor performance in these topics is the unbalanced dataset, the similarity in label classifications, and the distance of the RGB color encoding.

Results from the prediction dataset, which the network had never seen before, are shown in Figure 7(c). For unseen data, the limitations of the network are further shown, such as the top left example failing to detect any of the water. However, the overall performance across the classes is positive. Table 3 shows the quantitative performance metrics for the prediction dataset. A majority of the classes are successful at detecting; however, there is a concern for false positives and false negatives. Overall, results are promising and can be used in path planning; however, more training and more data would improve the predictions significantly.

<table>
<thead>
<tr>
<th>Table 2 Prediction Dataset Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Accuracy (%)</td>
</tr>
<tr>
<td>Parks</td>
</tr>
<tr>
<td>78.9</td>
</tr>
<tr>
<td>IoU Scores</td>
</tr>
<tr>
<td>0.73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3 Training Dataset Performance Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Accuracy (%)</td>
</tr>
<tr>
<td>Parks</td>
</tr>
<tr>
<td>95.9</td>
</tr>
<tr>
<td>IoU Scores</td>
</tr>
<tr>
<td>0.96</td>
</tr>
</tbody>
</table>
Table 4  3-Channel Label Encoding

<table>
<thead>
<tr>
<th>Label</th>
<th>Parks</th>
<th>Water</th>
<th>Parking</th>
<th>Sidewalk</th>
<th>Roads</th>
<th>Buildings</th>
<th>Trees</th>
<th>Poles</th>
<th>Unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>[0,255]</td>
<td>[0,0,255]</td>
<td>[128,128,128]</td>
<td>[90,90,90]</td>
<td>[125,125,125]</td>
<td>[155,0,155]</td>
<td>[0,120,10]</td>
<td>[205,255,0]</td>
<td>[0,0,0]</td>
</tr>
</tbody>
</table>

Fig. 7  Pix2pix Urban Landcover Predictions
B. Costmap Formation

Data layers of population density, landcover classification, and terrain slope were aligned, filtered, normalized with a range of zero and one, and aggregated to form a single costmap. More specifically, the costmap was generated as a linear combination of each layer multiplied by weightings. This work produced multiple costmaps for each of three scenarios by adjusting the weightings to explore how much costmap would fluctuate with variations in the weightings and how they would impact path planning. The weightings could be varied randomly, but this paper used Latin Hypercube sampling (LHS), one of the most popular design of experiments, to effectively spread values of weightings between zero and one. A total of 20 lists of weightings were created, as delineated in Table 5, and utilized for generating 20 distinct costmaps for each of the three scenarios. Examples of costmaps generated for three scenarios are illustrated in Figure 8.

<table>
<thead>
<tr>
<th>Parks</th>
<th>Water</th>
<th>Parkings</th>
<th>Sidewalks</th>
<th>Roads</th>
<th>Trees</th>
<th>Poles</th>
<th>Population</th>
<th>Terrain Slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3248</td>
<td>0.7528</td>
<td>0.2797</td>
<td>0.3775</td>
<td>0.7713</td>
<td>0.2742</td>
<td>0.4502</td>
<td>0.6248</td>
</tr>
<tr>
<td>2</td>
<td>0.6677</td>
<td>0.5225</td>
<td>0.3667</td>
<td>0.2897</td>
<td>0.0837</td>
<td>0.1671</td>
<td>0.4223</td>
<td>0.2194</td>
</tr>
<tr>
<td>3</td>
<td>0.8346</td>
<td>0.0136</td>
<td>0.1533</td>
<td>0.9550</td>
<td>0.4772</td>
<td>0.7025</td>
<td>0.8242</td>
<td>0.3590</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>19</td>
<td>0.1701</td>
<td>0.6673</td>
<td>0.6426</td>
<td>0.2083</td>
<td>0.2302</td>
<td>0.4013</td>
<td>0.6051</td>
<td>0.9082</td>
</tr>
<tr>
<td>20</td>
<td>0.7558</td>
<td>0.9964</td>
<td>0.0550</td>
<td>0.6195</td>
<td>0.5801</td>
<td>0.8715</td>
<td>0.2203</td>
<td>0.0634</td>
</tr>
</tbody>
</table>

Table 5 20 Lists of weightings corresponding to each data layer

C. Test Scenarios

Let us consider the case where a particular client has access to a VTOL vehicle to transport them from the Ronald Reagan Washington National Airport to the Washington Nationals Stadium. In this scenario, the Nationals have added a Vertiport or landing pad near Garage C near the stadium.

The alternative paths through the environment are formed using a path planning algorithm, as detailed in the background section. First, two classic algorithms are compared for their method of applying the costmap, the performance of the final path, and the computational requirements to run the calculation. Next, a risk-aware RRT algorithm is used to maximize the use of the costmap.

Next, the energy-aware RRT algorithm is used to produce trajectories. The RRT algorithm is modified to include potential energy metrics from the costmap and kinetic energy metrics from a linear dynamics approximation. The exploration of the algorithm in the costmap is shown with a 2D visualization in Figure 9.

Overall the energy-aware RRT algorithm produces unique trajectories from the larger amount of parameters that are chosen; however, the computational time is significantly longer when dealing with large search areas. Therefore, the other two scenarios do not examine this algorithm further, as additional parameter tuning is required.

An additional two scenarios are examined for further evaluation of the two vanilla algorithms and the change in costmap. The first experiments are shown in Figures 10(b) and 10. The evidence suggests that algorithms produce dramatically different results that each is largely dependent on the costmap.

D. Performance Considerations

The costmap formation process requires multiple steps to form, and then combine, the data layers into a riskmap. Each layer is unique in its creation and in this work it is assumed that the data preprocessing has been completed before preflight planning. Combining the data has only a few computational requirements; it only requires reprojecting the data to the same frame of reference and resolution, and then normalizing and adding the layers using matrix math. Modern geospatial and mathematical libraries can do these steps in seconds.

Previous work in discovering landing zones uses visual imagery at a resolution of 3-inch squares rather than the 1-meter for this work. The tradeoff is in the amount of data and the fidelity of the predictions. In addition, the higher resolution data can detect more precise patterns to use in training. However, after running tests, the 1-meter resolution showed successful results with a much faster training time. This may be related to the fact that the critical features in the image, such as lines and textures exist at the scale of structures such as street corners and rooftops.

The vanilla A* algorithm produces completed paths in a concise amount of time. For instance, in scenario 1, after the data has been preprocessed, the path is formed in less than five seconds on average. The vanilla RRT algorithm completes paths in a similar time, on average less than ten seconds. However, the energy-aware RRT algorithm runs for
Fig. 8 Example of Costmaps Generated for Three Scenarios
E. Discussion

Additional investigation is needed for improving the framework for the real-world applications pilots and UAM operators will encounter. For example, the alternative routing may need to be applied in flight, requiring faster processing and more efficient algorithms. In addition, the alternative landing sites may need to be formed preflight with geometric and dynamic landing constraints that provide higher confidence for safe emergency landing. This is seen as a critical capability by research into UAM, for example, Sanjiv Singh’s outline for Assured Autonomy, including a contingency planning module.

Further work can improve the risk-aware aspect of the planning by leveraging work by Sharma et al in [22]. The use of the mean-variance and Conditional-Value-at-Risk metrics for risk provides quantifiable safer paths. The integration of the additional datasets will require the collection or assumption of prior distributions for independent and conditional random variables, though, otherwise will result in biased predictions. In addition, the alternative routing approach used in [19] provides multiple potential paths within the tree structure. Future work should look to leverage this when having dynamic or noisy data layers.

A few techniques could improve the semantic segmentation stage. One, the resolution of the imagery could be increased. The current dataset is available at a 3-inch resolution; however, a 1-meter resampling is used in the assumption that lower resolution data is more widely available. Two, the amount of data could be increased by incorporating additional data. For instance, L.A. county provides many of the same data layers that were leveraged in this work. Lastly, the model architecture and training hyperparameters could benefit from parameter optimization to determine if training results for the semantic predictions can be improved. Harris et al, in [7], further evaluates the region with feasible landing zones based on distance and size. Further work could integrate energy metrics, as in [23], to properly decide what spots are feasible to reach. This may require dynamic and geometric constraints as in [24].

VI. Conclusions

The future of UAM and AAM operations in urban regions requires an increased level of safety and contingency planning. Therefore, a preflight planning tool is developed to provide VTOL pilots or operators with insight on flight paths through a region. A diverse set of data is combined and used for a risk minimized path. The resulting path can be modified by selecting new target locations, or multiple alternative routes can be found. Three test scenarios demonstrate the capability of this methodology in the Washington DC area. The Pix2pix GAN is trained to predict urban landcover successfully and then used as a layer within the costmap. The costmap is formed from the combination of multiple data layers, including the urban landcover and population density. A linear combination of data layers is aligned and normalized to produce a set of 2-dimensional costmaps. The vanilla A* and RRT and the energy-aware
(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Fig. 10 A* vs. RRT path planning
RRT* algorithms are compared for the speed and performance of the final path. Additional work is still necessary to validate the methodology and to improve the results from community feedback; however, the hope is that this tool will improve safety for future urban VTOL operations.

References


