Porting Computer Vision Models to the Edge for Smart City Applications: Enabling Autonomous Vision-Based Power Line Inspection at the Smart Grid Edge for Unmanned Aerial Vehicles (UAVs)

Ingi Steinn Gudmundsson
ETH Zurich
igudmunds@student.ethz.ch

Gregory Falco
Johns Hopkins University
University of Iceland
falco@jhu.edu

Abstract

Smart grid infrastructure must be monitored and inspected - especially when subject to harsh operating conditions in extreme, remote environments such as the highlands of Iceland. Current methods for monitoring such critical infrastructure includes manual inspection, static video analysis (where connectivity is available) and unmanned aerial vehicle (UAV) inspection. UAVs offer certain inspection efficiencies; however, challenges persist given the time and UAV operator skill required. Collaborating with Landsnet, the Icelandic smart grid operator, we apply convolutional neural networks for image processing to detect smart grid transmission infrastructure and modify the resulting computer vision (CV) model to function on the edge of a UAV. In doing so, we overcome significant edge processing barriers. Our real-time CV model delivers decision insight on the UAV edge and enables autonomous flight path planning for use in smart grid inspection. Our approach is transferable to other smart city applications that could benefit from edge-based monitoring and inspection.

1. Introduction

In this paper we explore opportunities for smart city asset monitoring using edge processing. Particularly, we focus on the potential to monitor energy infrastructure that includes smart grid components. Smart grids make use of sensors to obtain information about the status of the grid. They enable automation to ensure reliability and availability of the system. Visual inspections play a key role in order to assess the status of power grids. These inspections are usually carried out by operators on foot or by helicopter. Current inspection methods face several challenges as they can be time-consuming, dangerous for operators and expensive. These challenges are a function of the immense area the system covers, their remote locations, hostile operating environments and the vast amounts of complex components requiring inspection. Such challenges are not exclusive to electric infrastructure, but to other smart city critical infrastructure elements such as bridges, tunnels and building inspections.

Certain geographies are more prone to inspection challenges than others. Our study is being conducted in partnership with Landsnet, the smart grid operator in Iceland. Iceland is an island nation in the middle of the Atlantic ocean that is subject to persistently high winds and extreme weather. Energy infrastructure is expensive and difficult to maintain in this environment which prompts the need for automation of monitoring solutions. Some regions of Iceland are largely uninhabited and take hours to drive to over difficult terrain; however, smart grid infrastructure passes through these regions and they require the same extent of monitoring and maintenance as grid components in populated areas.

Unmanned aerial vehicles (UAVs), also called drones, have been proposed as a solution to remote monitoring problems as they offer a comparatively low-cost and convenient means for monitoring remote infrastructure assets[1]. UAVs can be controlled by operators to fly along power lines and capture images of the smart grid infrastructure for further analysis. While this provides a means to collect data about the state of the smart grid, it remains a relatively time and human resource intensive process. Drone operators generally collect image data without insight to where defects may be found. This results in the collection of a significant amount of video footage that requires post-processing and analysis manually by an operator. Essentially, trained technicians need to watch (hours of) video footage taken by the UAV to detect subtle issues with the transmission infrastructure. Communication connectivity limitations in remote regions where such smart grid infrastructure is found renders live data streams back to an operations center infeasible. Parallels can be made to other smart city assets that are difficult to reach such as conducting bridge inspections over bodies of water. Generally image data must be first collected and stored on the edge and then physically brought back

URI: https://hdl.handle.net/10125/80271
978-0-9981331-5-7

for the manual analysis described. This prompts the question: can real-time edge-based image processing be used to facilitate autonomous UAV path planning that enables remote smart grid infrastructure inspections?

For a fully automated inspection where the UAV could autonomously fly along the inspection path (the power line) and then detect specific problem areas for the smart grid in real-time, optimized algorithms and models will be required for edge processing. Convolutional neural networks (CNNs), a machine learning technique and form of artificial intelligence (AI), have demonstrated strong results in image processing but often require an extensive amount of computing resources making them hard to use in resource constrained environments such as UAVs. One option to address the edge processing constraint is to engage cloud based processing where the majority of computations would be offloaded to remote computing clusters and the results could be sent back to the UAV, avoiding the resource constraints of the UAV. However, due to the often remote locations of infrastructure, where there is often limited or no connectivity, an edge processing solution could be preferable. Efficient neural networks have been proposed that could potentially help serve the remote monitoring needs on the edge of a UAV[2, 3]. In this paper we develop a CNN capable of running onboard a UAV at the edge for real-time decision insights such as path planning and smart grid defect monitoring.

2. Prior Art

As transmission infrastructure ages and is tasked with demands beyond their design parameters due to increasing energy consumption, they are prone to failures. Given the critical nature of energy infrastructure and our society's reliance on its 24/7 operations, rapid detection of issues relating to transmission systems such as downed wires, fraying electric cables or other anomalies is essential to enabling fluid electric grid service. Degradation of energy infrastructure becomes more common in the wake of extreme weather and natural disasters, which have been more severe and frequent in recent years due to climate change [4].

Monitoring of transmission infrastructure has been conducted in numerous ways. The most common methods have remained unchanged for decades and include visual surveys by foot and helicopter-assisted inspection. Visual inspection by foot is carried out by a team walking along the transmission line and inspecting both the lines and towers with the help of binoculars, and various cameras such as infrared and corona detecting cameras. This method is time-consuming, costly and dangerous for the maintainers but can provide a higher detection rate than other methods. The helicopter-assisted method is faster and requires less labor-intensive work but generally has a lower detection rate. The helicopter is flown along the line and a maintainer visually inspects the line and gathers data with cameras for further analysis. Even though this method requires less manpower and time it is more expensive compared to visual surveys by foot [5, 6].

With the introduction of smart grid data collection techniques different strategies have been suggested to monitor the status of the grid. Sensor networks have been deployed to measure and monitor the health of the system [7]. Different sensors such as temperature and tilt sensors are placed on various assets of the grid to detect mechanical failures and provide a complete physical and electrical status of the system in real-time. However, these networks come with challenges. One of the main drawbacks of this approach is that the sensors are static and thus the energy grids require a vast amount of sensors placed along the system to properly monitor it [8]. Further, in many cases there is need for communication systems throughout the smart grid to relay the data back to operators that analyze the data[9]. Such remote communication systems, known as mesh networks and offered by providers such as Silver Springs Network, have been specifically developed for smart grid infrastructure, but are not sufficient for high throughput data transfer such as video data[10, 11].

With the advancements in recent years in deep learning, namely, convolutional neural networks(CNN), computer vision(CV) methods for visual inspection have greatly improved. Previous work on the inspection of smart city structures has focused on developing CV models that require significant compute and memory allocation. CV models are employed for a variety of smart city applications ranging from CCTV facial recognition for suspect identification to roadside occupation detection for intelligent transport services[12, 13]. However, challenges persist in bringing these models to be performant on the edge, enabling real-time decision insight. In most cases, fiber optic communications links are needed for video feeds to be communicated to locations where the data can be processed. This limitation has presented challenges for UAVs to conduct deep inspection of smart city infrastructure [14].

Despite this limitation, in recent years UAVs have gained attention as a potentially disruptive technology to enable visual inspection of transmission systems[15]. Visual inspection for other infrastructure such as buildings has been engaged for technical inspection[16].

UAVs are relatively low-cost, and easy to deploy - with a primary cost being a skilled UAV operator. UAVs can fly closer to the components of the transmission system than helicopters resulting in more detailed images of the components. They are typically equipped with various sensors and cameras to gather data about the status of the components and for navigation. UAVs have been used for mapping and inspection of transmission lines as well as vegetation encroachment monitoring near transmission lines[17]. However, no fully automatic autonomous vision-based inspection system that is capable of detecting a wide range of faults has been successfully developed but many task-specific approaches exist for navigation and object detection [1].

A promising example of CV for smart grid monitoring was developed by Nguyen et. al and is called LS-Net where power lines were identified in energy infrastructure images, which can inform operators of transmission issues [18]. The intention was to identify the transmission lines for indicators of deficiency and so that the lines could inform a path ultimately fed to a UAV for autonomous flight planning purposes. The authors reported that it can detect power lines from images in real-time on an NVIDIA TITAN X GPU. LS-Net is a CNN inspired by single-shot object detectors like SSD and YOLO. It consists of three modules, a fully convolutional feature extractor based on VGG-16, a classifier and a line segment regressor. Instead of predicting the pixel level segmentation of the power lines LS-Net splits the input image into four overlapping grids and for each cell in the grid, it predicts the probability that the cell contains a power line segment and the coordinates of the endpoints of the longest line segment in each cell. While the team's ambition was to feed power line image detection into path planning waypoints for a UAV flight controller, LS-NET is too processing intensive and as it currently stands and is inoperable on edge GPUs.

Several approaches exist to compress neural networks for edge applications. These include, weight sharing, pruning, quantization, tensor decomposition, and knowledge distillation[19]. Using these methods, CV models have been demonstrated to operate on UAVs, which could be informative for modifying the LS-Net model so that it can work onboard a UAV. These CV models specifically focus on fault identification on the edge of the UAV. Medrano et al uses -Mobilenet v1, a single shot detector CNN with Mobilenet as a feature extractor to detect power masts in images[20]. They propose an algorithm to use this detector to point a camera gimbal at the closest tower to maximize the coverage of the tower for data collection based on GPS coordinates of the UAV and the power mast. This is done in real-time and on-board the UAV using a TPU, the Google Coral USB Accelerator. Another team that aimed to accomplish fault detection on the edge includes Ayoub et al. They use the YOLOv3 and YOLOv3-tiny models for real-time on-board detection of power line components and faults[21]. They deploy these models on 4 different single board devices(SBDs) that can be embedded as part of an UAV i.e. Raspberry Pi 4, Nvidia Jetson Nano, Nvidia Jetson TX2 and Nvidia AGX Xavier. They then evaluate their real-time performance on-board. They also use TensorRT from NVIDIA, which is a deep learning inference optimizer, to optimize the weights of the network after training and compare the results. Leveraging Medrano and Ayoub's teams work on developing efficient CV to be performant on edge devices such as the Jetson and Coral, we aim to enable LS-Net's features on the edge of the UAV as originally intended by Nyugen et al. This can ultimately yield real-time path planning waypoints to enable fully autonomous inspections of smart grid infrastructure using UAVs.

3. Our Approach for Edge CV Adaptation

The LS-Net model proposed by Nguyen served as a base for our edge modification work. Since Nguyen and team did not provide their implementation of the network we implemented it using TensorFlow [22] and Keras [23].

3.1. Data preprocessing

The training dataset we used in our experiments is the newly released TTPLA dataset [24], which contains aerial images of both transmission towers and power lines and their pixel-level annotations for instance segmentation in COCO format [25]. Since the output of LS-Net is not the pixel-level annotations of the power lines as described above, data preprocessing was necessary. Additionally, we used the Ground Truth of Powerline Dataset (Infrared-IR and Visible Light-VL) [26] to further assess the robustness of the replicated LS-Net model.

Our first step involved resizing the images to 512x512 to match the input size of the network. Then we created labels from the segmentations of the images. As previously mentioned, each image is split into four overlapping grids. Each grid cell in these four grids must include the class label, stating whether or not the cell contains a line segment, and two coordinates corresponding to the endpoints of the longest line segment. Getting the class labels for the segmentations is a trivial task where we just have to check if any pixel in the cell is a line segment. However, collecting the endpoints is more challenging. We achieved this by taking each line segment in each cell, and fitting a line to all the points that comprise the segment using a weighted least-squares method. Then we create a convex hull containing all the points on the line, extract the points that form the convex hull and finally calculate the Euclidian distance between each pair of points and return the pair that is farthest apart. Calculating the distance between all the points on a line is an expensive task taking in general $O(n^2)$ time. Using the fact that the two farthest points lie on the convex hull and that we can obtain the convex hull in $O(nlogn)$ time and extract the points that form the convex hull we greatly reduce the number of distance calculations we have to perform since the number of points on the convex hull is generally significantly less than all the points on the line.

3.2. Model Characterization

In order to better understand the requirements and measure the performance of the network we used several benchmarks. Before we started the process of porting the CNN to the edge we wanted to better understand the computational cost, memory requirements and the throughput of the network in frames per second (FPS).

Computational Cost: A common approach to calculate the computational cost of a CNN is to count the number operations performed during inference. Multiply–accumulate operations (MACs) are the most common operations during inference in convolutional neural networks [27] and thus provide a sufficient estimate of the computational cost. The formula we used to count the number of MACs is:

$$MACs = K_0 * K_1 * C_{in} * H_{out} * W_{out} * C_{out} \quad (1)$$

where K_0 and K_1 are the kernel width and height, C_{in} is the number of input channels, H_{out} and W_{out} are the height and width of the output feature map of the layer and C_{out} is the number of output channels of the layer.

Memory requirements: Another important metric to consider in resource constrained environments is the total memory required for inference. In general the less memory the model takes the faster it can run. To assess the memory requirements of the models, we measured the memory required to store the weights of the model and the two largest consecutive feature maps. In doing so, we further accounted for the CNN precision since lower precision networks require less memory to store their parameters.

Throughput: One of the most important characterization metrics for computer vision is the model throughput, measured in frames per-second (FPS), of the network during inference. In order to make decisions in real time for the UAV the model has to be able to process its input in real time. We measured the FPS by dividing the batch size by the average execution time of 1000 batched inputs after warming up the GPU for 100 inputs.

Evaluation metrics: To compare the performance of the models and assess the effect of different shrinking methods we calculated the pixel-level average recall rate (ARR), precision rate (APR) and F1 score. Since LS-Net does not output the pixel-level segmentation maps we replicated the evaluation metrics used for LS-Net for comparative purposes [18]. First, we apply the 8-connected Bresenham algorithm to approximate a straight line between the two endpoint predictions. Then we smooth the segmentation by convolving a 2D Gaussian kernel over the maps. Finally, we use Otsu's method for thresholding to collect the binary segmentation maps.

3.3. Porting CV to the Edge

To test our model on the edge we selected an edge device that was capable of running our model, while still being small enough to fit on an UAV. The Nvidia Jetson Nano and the Google Coral Dev Board are small and relatively inexpensive single-board computers that can easily be mounted to the UAV for inference at the edge. The Jetson Nano price is 99$ while the Coral is slightly more expensive, costing roughly 169$.

The *Nvidia Jetson Nano* is a small single-board computer. It is equipped with a Quad-core ARM A57 processor, 128-core NVIDIA Maxwell™ GPU capable of 472 GFLOPs using half precision(FP16) and has 4 GB of 64-bit LPDDR4 memory[28].

The *Google Coral Dev board* is, like the Jetson Nano, a single-board computer. It is equipped with a i.MX8M system-on-chip which includes the ARM Cortex-A53 processor, ARM Cortex-M4 processor, Integrated GC7000 Lite Graphics GPU and 4 GB of LPDDR4 SDRAM memory. However, unlike the Jetson Nano the Google Coral Dev Board includes the Edge TPU coprocessor for ML inference capable of 4 trillion operations per second(TOPS) using 8 bit integer(INT8) precision[29].

These devices have significant resource constraints since they are built to be compact, efficient and inexpensive. Additionally, they have limited support for operations and precision. We ultimately selected the Jetson Nano for our primary test system because it provided slightly more flexibility for our model since the Coral only supports INT8 precision. To speed up inference on these devices the model needs to be optimized for the system it is running on. To optimize our network for the Jetson Nano we used TensorRT.

TensorRT is a high-performance neural network inference optimizer and runtime engine. TensorRT can optimize neural networks by combining layers and optimizing kernel selection for improved latency, throughput, power efficiency and memory consumption. Additionally, it supports quantization to run inference at lower precision. The Jetson Nano supports half-precision floating point(FP16) numbers so we used TensorRT to optimize the network for the Jetson Nano and to quantize it to FP16 for increased performance [30].

4. Results

Below we detail the outcomes from our effort to modify the LS-Net model to be suitable on the Jetson Nano edge device.

4.1. LS-Net

LS-Net is a fully convolutional neural network that consists of 16 2D Convolutional layers followed by a Group Normalization layer and a ReLU activation. However, given TensorRT's limited support we swapped the Group Normalization for the more commonly used Batch Normalization layer. Table 1 shows the high level description of the architecture of the LS-Net. Each Convolutional layer in the feature extractor is padded to maintain the size of the input.

Input ($512 \times 512 \times 3$)	
Conv $3 \times 3 \times 64$	
Conv $3 \times 3 \times 64$	
Conv $3 \times 3 \times 64$ S2	
Conv $3 \times 3 \times 128$	
Conv $3 \times 3 \times 128$	
Conv $3 \times 3 \times 128$ S2	
Conv $3 \times 3 \times 256$	
Conv $3 \times 3 \times 256$	
Conv $3 \times 3 \times 256$ S2	
Conv $3 \times 3 \times 512$	
Conv $3 \times 3 \times 512$	
Conv $3 \times 3 \times 512$ S2	
Conv $2 \times 2 \times 512$	Conv $2 \times 2 \times 512$
Conv $1 \times 1 \times 2$	Conv $1 \times 1 \times 4$

Table 1. LS-Net architecture

4.2. Model Modification

Since the Jetson Nano has limited capabilities we sought to compress the model to increase its throughput. We ended up with two variants of the original LS-Net that we call Tiny LS-Net and Tinier LS-Net. Table 2 shows the high level architecture of the Tiny LS-Net. The main difference between the original and the tiny version is that we utilize 1×1 convolutions to reduce the number of channels in the models that resultantly reduces the computational complexity.

Input (512 x 512 x 3)	
Conv $3 \times 3 \times 64$	
Conv $3 \times 3 \times 64$ S2	
Conv $3 \times 3 \times 128$	
Conv $1 \times 1 \times 64$	
Conv $3 \times 3 \times 128$ S2	
Conv $3 \times 3 \times 256$	
Conv $1 \times 1 \times 128$	
Conv $3 \times 3 \times 256$ S2	
Conv $3 \times 3 \times 512$	
Conv $1 \times 1 \times 256$	
Conv $3 \times 3 \times 512$ S2	
Conv $1 \times 1 \times 256$	Conv $1 \times 1 \times 256$
Conv $2 \times 2 \times 512$	Conv $2 \times 2 \times 512$
Conv $1 \times 1 \times 2$	Conv $1 \times 1 \times 4$

Table 2. Tiny LS-Net architecture

To reduce the size of the model even further we also tried halving the number of channels in every convolutional layer except for the last layer. This resulted in the tinier variant which has a much greater throughput than both the previous versions.

Input (512 x 512 x 3)	
Conv $3 \times 3 \times 32$	
Conv $3 \times 3 \times 32$ S2	
Conv $3 \times 3 \times 64$	
Conv $1 \times 1 \times 32$	
Conv $3 \times 3 \times 64$ S2	
Conv $3 \times 3 \times 128$	
Conv $1 \times 1 \times 64$	
Conv $3 \times 3 \times 128$ S2	
Conv $3 \times 3 \times 256$	
Conv $1 \times 1 \times 128$	
Conv $3 \times 3 \times 256$ S2	
Conv $1 \times 1 \times 128$	Conv $1 \times 1 \times 128$
Conv $2 \times 2 \times 256$	Conv $2 \times 2 \times 256$
Conv $1 \times 1 \times 2$	Conv $1 \times 1 \times 4$

Table 3. Tinier LS-Net architecture

4.3. Model Characterization

Table 4 depicts the benchmark results for the three different variants. The number of parameters, number of MACs, memory required for the weights and the FPS using FP16 precision are noted. By using the 1×1 convolutions the tiny version has less than half the number of parameters and MACs and requires roughly 40% less memory compared to the original model. Likewise, the tinier version has only about $1/10$ of the parameters and MACs and requires around 75% less memory compared to the original. However, compared to the original the tiny variant has 3x higher throughput and the tinier variant has 9x higher throughput. In addition, Nguyen et al [18] reported a throughput of 20.4 FPS on a GeForce 1080 Ti GPU which is a much more powerful GPU than the Jetson Nano. Given that the tinier variant runs at 27 FPS on the Jetson Nano we have obtained a significantly faster model thanks to the efficiencies afforded by our modifications.

The consequence of shrinking the model is that the accuracy will be affected. Table 5 shows the ARR, APR and F1 scores for each variant on both datasets. The goal of this project is not to obtain the most accurate model but to have a model that is reasonably capable of detecting power lines while still being able to run on the edge at a speed necessary to enable real-time decision insight for path planning. As can be seen from table 5 the ARR, APR and F1 scores are in most cases lower for the smaller models. However, given the improvement in throughput this difference is justifiable. Furthermore, visually comparing the model shows that even though the models have less accuracy they are still able to properly detect the smart grid infrastructure. However, as can be seen in figure 1 the results get noisier with the reduced model size. It is worth noting that as expected the scores are marginally lower on the TTPLA dataset, compared to the Mendeley dataset, due to the more complex and cluttered environment in those images i.e. fences and buildings. Such features degrade the prediction accuracy. The Mendeley dataset mostly contains images of power lines with a simple background of grass and trees. While simpler, the Mendeley dataset matches more closely to the background expected in Iceland where this research is conducted.

To further visually assess the performance of the model we obtained image data from Landsnet, the Icelandic smart grid operator. As can be seen from figure 2 the model is capable of detecting transmission lines from these images as well, even though the surroundings differ from the training set. This is further discussed below in our limitations section.

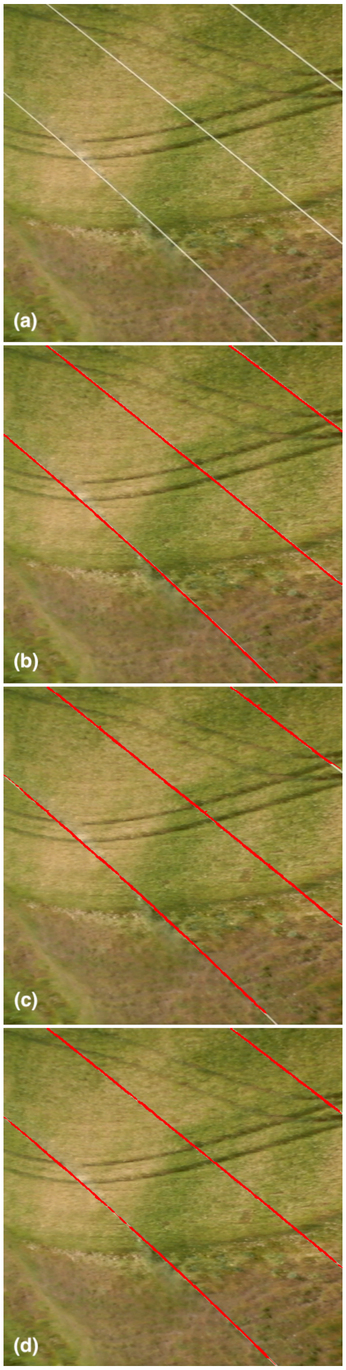

Figure 1. Visual comparison of the three variants on the Ground Truth of Powerline Dataset (Infrared-IR and Visible Light-VL). (a) Original image, (b) LS-Net BN, (c) Tiny LS-Net BN, (d) Tinier LS-Net BN

5. Discussion

Demonstrating computer vision functionality on the Jetson Nano is pivotal to generating decision insight

	#Parameters	MACs	Memory (FP16)	FPS (FP16)
LS-Net BN	9,934,598	65 G	82.9 MB	3
Tiny LS-Net BN	4,634,630	24 G	48.8 MB	9
Tinier LS-Net BN	1,163,270	6 G	22.2 MB	27

Table 4. Benhmark results for the three different variants of LS-Net.

	TTPLA			Mendeley Dataset		
Model	**ARR**	**APR**	**F1 score**	**ARR**	**APR**	**F1 score**
LS-Net BN	**0.6022**	**0.3786**	**0.4276**	**0.7593**	**0.4749**	**0.5351**
Tiny LS-Net BN	0.5907	0.3587	0.4128	0.7302	0.4527	0.5083
Tinier LS-Net BN	0.5611	0.3353	0.3861	0.7119	0.4355	0.4917

Table 5. Pixel-level average recall rate(ARR), precision rate(APR) and F1 score evaluated on TTPLA [24] and Ground Truth of Powerline Dataset (Infrared-IR and Visible Light-VL)(Mendeley Dataset) [26].

on the edge of a smart grid monitoring system such as a UAV. To date, we successfully mounted the Jetson Nano to our self-assembled UAV, which we have tested and successfully processed live video feeds in real-time. There are multiple use cases for this demonstrated capability, which helps to guide our future work. In addition to describing how we plan to build on these results, we discuss some of the generalizable benefits of our work and the limitations of our study.

5.1. Future Work

First, image processing at the edge of a drone that can successfully detect power lines in remote regions can be engaged for path planning and degrees of autonomous operations for the UAV. Today, UAVs engaged for smart grid monitoring and analysis must be operated by a trained pilot - especially given the safety-critical nature of the work (UAVs must avoid active transmission lines). Further, any automation is dependent on GPS navigation, which can be unreliable in certain geographies with signal interference (such as mountains or tall buildings). This presents considerable limitations for remote energy infrastructure such as our test locations in regions of Iceland. Additionally, while not the foremost concern for energy infrastructure inspection, GPS navigation can also be spoofed or jammed by an attacker. Engaging edge processing on the Jetson Nano to inform path planning using a CNN without the use of GPS could provide an alternative or complementary means to controlling the drone in low visibility and harsh terrain where operators have less ability to control or physically follow the drone to facilitate safe flight operations. One means for autonomous flight path planning includes engaging the edge CV model to control the UAV's direction based on the CV model predictions that could be fed into a drone's flight controller - also located on the edge of the UAV. The CV model could iteratively provide such direction based on the transmission lines in focus to collect relevant video data for future inspection by trained operators. Such a use case will be an evolutionary improvement to existing means of remote smart grid monitoring as a highly trained human pilot would no longer become critical path to capturing the relevant inspection video data. We intend to develop this functionality over the coming months by integrating the Jetson Nano to the flight controller on our UAV.

A second and more revolutionary use case for the Jetson Nano's CV capabilities on the UAV includes the integration of path planning autonomy and automated defect inspection for smart grid infrastructure - the elusive application described earlier. Such a capability will first require the aformentioned autonomy use case. For defect inspection, an additional CV model will be needed, such as those described by Medrano et al and Ayoub et al to detect abnormalities in the smart grid infrastructure[20, 21]. We would need to evaluate if both CV models could effectively function on the same Jetson Nano or if an additional GPU would be required. Next, we would integrate the defect detection model with the current line detection model so that rather than waypoints be determined exclusively based on the location of the lines, they would be assigned based on the location of the lines and, where relevant, interim focus points where defects were detected. Such integration of anomaly detection with autonomous path planning will reduce the burden on smart grid inspectors by allowing them to focus their time investigating the defects that the UAV autonomously flew to, focused on and captured in a video feed, rather than spending time manually flying the drone along the transmission lines and then retrospectively finding them in video feeds. While a more ambitious target, we aim to establish this capability in partnership with Landsnet to inspect their remote smart grid infrastructure.

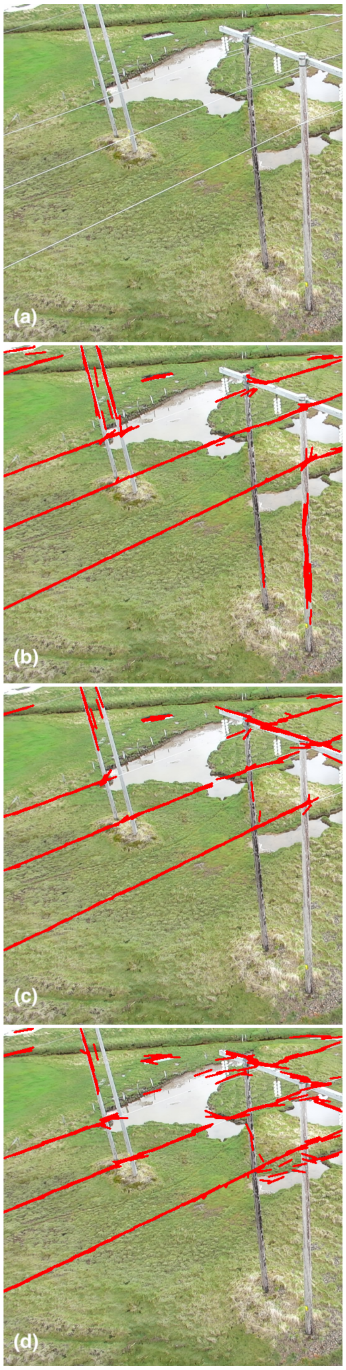

Figure 2. Visual comparison of the three variants on images from Landsnet. (a) Original image, (b) LS-Net BN, (c) Tiny LS-Net BN, (d) Tinier LS-Net BN

Finally, there are opportunities to continue condensing the CV model in order to enable multiple models to run on the same Jetson Nano. This could be especially useful for the fully autonomous inspection use case described above. An emergent technique to condense deep learning models is to engage transformers. Transformers have been proposed for use in remote sensing image classification, which could be applicable for our study [31]. We hope to experiment with this technique as our research continues.

5.2. Generalizable Benefits

There are significant benefits afforded by enabling real-time decision insight on edge systems such as UAVs. While we have been studying this in the context of enabling smart grid infrastructure monitoring, other smart city use cases could equally benefit from autonomous flight planning and inspection. For example, first responders such as firefighters increasingly engage UAVs to aid in situational awareness. However, their use of UAVs is primarily limited to video capture. With edge-based CV models, UAVs could help to deliver decision insight to first responders. For example, a UAV may be able to follow a firefighter into a smokey area and engage optical sensors such as an infrared camera coupled with edge CV to detect a child in a building. In this case, a firefighter or operator at a command center does not have to look through video footage to discern this, instead the CV model could automatically detect and inform the firefighter of the child accordingly. Similar scenarios can be envisioned for law enforcement officers as well as they are dealing with matters such as hostage situations. The advancements described concerning our methods to compress our edge CV model could be applied for these other applications with limited modifications.

5.3. Study Limitations

While the success of porting the CV model to the Jetson Nano is encouraging, there are limitations to our study that should be more closely investigated. First, the images used to train the model were simple, lacking dynamic backgrounds with confusing landscapes that may otherwise impede outcomes. For example, the image depicted above of the Icelandic power line has a line segment that is not detected by the CV model. This is likely because the model is not robust enough given its training set to work on images with landscape features as the one in this Icelandic landscape image. In the future, we aim to retrain the model on labeled Icelandic power line data, which will be considerably more dynamic in nature. To our current model's credit, it is difficult to see the transmission line with the naked eye for the Icelandic image; however a better fit line could perhaps be enabled by training the model on the Icelandic landscape. Similarly, we have not tested the CV model with considerable image interference as could be caused by extreme weather events. It is likely that the model would perform in a less accurate or robust manner should the images be occluded. Equally, we can try to address this by augmenting the training data to include such occluded images. These limitations point to the need for rigorous stress-testing of these models - especially if they are to be used in the context of the future use cases described above for smart grid infrastructure monitoring in remote, extreme environments.

A second limitation of our study is that we have not calculated the accuracy of the individual CV model tasks that are used in aggregate for transmission line prediction. Currently, the model predicts if a line is present and then establishes the endpoints for the line in the frame. The accuracy measure we ultimately use relates to closeness of the simulated line generated based on the aggregate of the two tasks. This fails to account for the accuracy of the individual tasks (line prediction and endpoint prediction). While there are likely few practical deficiencies as a result of our current approach, should we seek to refine the model's accuracy further, it would be useful to optimize around the accuracy for the individual tasks rather than the aggregate of the two. Further study would also benefit from establishing an audit trail in the form of log files to continuously monitor the predictive accuracy of the CV model to help govern the AI's operational safety[32].

A third limitation of our study is that we test the network on a singular smart city application - the smart grid. While our approach for shrinking the CNN to the edge should work for other smart city applications such as construction monitoring or crowd surveillance as well, these have not been tested. Additional challenges will likely arise should the UAV fly in a crowded cityscape that would complicate the image background and reduce the prediction accuracy of the model. Further, this study did not assess the policy constraints of flying a UAV in a crowded area that may have flight regulatory restrictions. Such an analysis should be conducted as part of future study.

A final limitation of the study is that we did not compare our approach to other possible methods to shrink the CV model. Our research highlights one potential means for porting a CV model to the edge, but there are likely other efficient methods of achieving this, including using transformers as described above. We aim to explore these options further as we continue to improve the accuracy and speed of our model on the Jetson Nano.

6. Conclusion

While smart grid monitoring systems are generally considered to include sensors that directly touch and engage with the transmission and distribution systems, mobile data collection and analysis solutions such as UAV monitoring and inspection technology will increasingly be considered core to what makes a smart grid ”smart”. This effort was the first step to enable our vision of autonomous path planning for smart grid infrastructure inspection. We aspire to help transform the monitoring and inspection landscape for smart city infrastructure by continuing our research efforts and integrating our compressed CV model with automated defect detection models, enabling autonomous vision-based power line inspection at the edge using UAVs. Our methodology proposed for porting CV models to constrained devices and thereby enabling real-time decision insight at the edge could facilitate use cases thought to be infeasible due to their processing requirements for the broader smart city community.

Acknowledgements

The authors would like to thank Landsnet for their research cooperation. The research was funded by the National Science Foundation, the Icelandic Fulbright Commission and Swiss Re.

References

[1] V. N. Nguyen, R. Jenssen, and D. Roverso, “Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning,” *International Journal of Electrical Power Energy Systems*, vol. 99, pp. 107–120, 2018.

[2] J. Wang, Z. Feng, Z. Chen, S. A. George, M. Bala, P. Pillai, S.-W. Yang, and M. Satyanarayanan, “Edge-based live video analytics for drones,” *IEEE Internet Computing*, vol. 23, no. 4, pp. 27–34, 2019.

[3] J. Lin, W.-M. Chen, Y. Lin, J. Cohn, C. Gan, and S. Han, “Mcunet: Tiny deep learning on iot devices,” *arXiv preprint arXiv:2007.10319*, 2020.

[4] U. D. of Energy, “Reimagining and rebuilding america's energy grid,” Jun 2021.

[5] L. Matikainen, M. Lehtomäki, E. Ahokas, J. Hyyppä, M. Karjalainen, A. Jaakkola, A. Kukko, and T. Heinonen, “Remote sensing methods for power line corridor surveys,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 10–31, 2016.

[6] J. Katrasnik, F. Pernus, and B. Likar, “A survey of mobile robots for distribution power line inspection,” *IEEE Transactions on Power Delivery*, vol. 25, no. 1, pp. 485–493, 2010.

[7] R. Leon, V. Vittal, and G. Manimaran, “Application of sensor network for secure electric energy infrastructure,” *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 1021–1028, 2007.

[8] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid — the new and improved power grid: A survey," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[9] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, "The applications of wifi-based wireless sensor network in internet of things and smart grid," in *2011 6th IEEE Conference on Industrial Electronics and Applications*, pp. 789–793, IEEE, 2011.

[10] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Analysis of communication schemes for advanced metering infrastructure (ami)," in *2014 IEEE PES General Meeting— Conference & Exposition*, pp. 1–5, IEEE, 2014.

[11] C. Bennett and D. Highfill, "Networking ami smart meters," in *2008 IEEE Energy 2030 Conference*, pp. 1–8, IEEE, 2008.

[12] V. A. Kumar, V. A. Kumar, S. Malathi, K. Vengatesan, and M. Ramakrishnan, "Facial recognition system for suspect identification using a surveillance camera," *Pattern Recognition and Image Analysis*, vol. 28, no. 3, pp. 410–420, 2018.

[13] G. T. S. Ho, Y. P. Tsang, C. H. Wu, W. H. Wong, and K. L. Choy, "A computer vision-based roadside occupation surveillance system for intelligent transport in smart cities," *Sensors*, vol. 19, no. 8, p. 1796, 2019.

[14] M. A. Ezzat, M. A. Abd El Ghany, S. Almotairi, and M. A.-M. Salem, "Horizontal review on video surveillance for smart cities: Edge devices, applications, datasets, and future trends," *Sensors*, vol. 21, no. 9, 2021.

[15] C. Kanellakis, E. Fresk, S. S. Mansouri, D. Kominiak, and G. Nikolakopoulos, "Autonomous visual inspection of large-scale infrastructures using aerial robots," *arXiv preprint arXiv:1901.05510*, 2019.

[16] A. Banaszek, S. Banaszek, and A. Cellmer, "Possibilities of use of uavs for technical inspection of buildings and constructions," in *IOP Conference Series: Earth and Environmental Science*, vol. 95, p. 032001, IOP Publishing, 2017.

[17] A. McFadyen, F. Dayoub, S. Martin, J. Ford, and P. Corke, "Assisted control for semi-autonomous power infrastructure inspection using aerial vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5719–5726, IEEE, 2018.

[18] V. N. Nguyen, R. Jenssen, and D. Roverso, "Ls-net: Fast single-shot line-segment detector," 2020.

[19] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[20] P. Medrano, J. Villadangos, and J. Astrain, "Uas: Iot on-line sensors for power line inspection," in *2020 IEEE Sensors*, pp. 1–4, IEEE, 2020.

[21] N. Ayoub and P. Schneider-Kamp, "Real-time on-board detection of components and faults in an autonomous uav system for power line inspection," in *1st International Conference on Deep Learning Theory and Applications, DeLTA 2020*, pp. 68–75, SCITEPRESS Digital Library, 2020.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[23] F. Chollet *et al.*, "Keras," 2015.

[24] R. Abdelfattah, X. Wang, and S. Wang, "Ttpla: An aerial-image dataset for detection and segmentation of transmission towers and power lines," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[25] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[26] N. Yetgin, Ömer Emre; Gerek, "Ground truth of powerline dataset (infrared-ir and visible light-vl)," 2019. Mendeley Data, V9, DOI: 10.17632/twxp8xccsw.9.

[27] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[28] NVIDIA, "Nvidia announces jetson nano," 2019.

[29] G. Coral, "Coral dev board data sheet," 2019.

[30] NVIDIA, "Tensorrt," 2021.

[31] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sensing*, vol. 13, no. 3, p. 516, 2021.

[32] G. Falco, B. Shneiderman, J. Badger, R. Carrier, A. Dahbura, D. Danks, M. Eling, A. Goodloe, J. Gupta, C. Hart, *et al.*, "Governing ai safety through independent audits," *Nature Machine Intelligence*, vol. 3, no. 7, pp. 566–571, 2021.