ECE 50024 Final Checkpoint Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications Paper Review

Team 20: Kyung Min Ko Kunal Mamtani Jonathan Stoschek Heron Teegarden Alec Vucsko

Code: GitHub

1. Introduction

Image restoration methodologies, such as Plug-and-Play Alternating Direction Method of Multipliers (PNP ADMM) presented in the paper entitled "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications" (Chan et al., 2016), offer algorithms that can be incorporated into real world technologies. The goal of the current work is to improve the methodology presented in this reference to apply it to automatic control of autonomous vehicles. To accomplish this goal, the PNP ADMM was reimplemented in Python, based on the methodology presented in the referenced paper. The mathematical basis of the PNP ADMM and how it improved upon the base ADMM was then analyzed to identify points of improvement upon the original ADMM method. Following this analysis, improvements were tested to verify that the implementation of the modified PNP ADMM is able to successfully de-blur an image. Finally, we addressed the specific problem, identification of traffic signs from low quality images in the autonomous control field, as these images may be very low resolution and have noise or blurring that disrupts the clarity of the image. With this implementation, the results demonstrate that the PNP ADMM can be useful in the automatic control of autonomous vehicles based on achieving specific improvements from the original ADMM algorithm.

2. Problem Statement

The paper "Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications", written by Chan et al, aims to introduce improvements to existing ADMM algorithms.

The improvements made to the existing ADMM algorithm include the ability to plug any bounded denoiser algorithm into the ADMM to more easily tailor the algorithms towards the application. This is done by replacing equation 2 with

$$v^{(k+1)} = D_\sigma\left(\tilde{v}^{(k)}\right). \tag{1}$$

The implementation of an adaptive update rule also increases the robustness. The ultimate goal of the paper is to prove the stability of the ADMM algorithm, which enables the use of any denoising algorithm in a defined subset of denoisers.

2.1. Plug-and-Play ADMM

There are two types of problems associated with current PNP ADMM. Existing methods of PNP ADMM image restoration is unclear about what conditions the denoiser must meet and what kind of denoising algorithms would it guarantee the convergence of the algorithm. Therefore, it is important to determine the necessary conditions and specific requirements for denoising algorithms to satisfy convergence. Moreover, it splits the variables in their own characterized method, so it is unclear to the user that is it possible to create a fast implementation for solving common Gaussian and Poissonian image restoration problems. As a result, it is crucial to obtain a fast algorithm for the inversion step.

2.2. Why PNP ADMM

The problem addressed by Chan et al. in this paper is important because previous ADMM algorithms (Boyd et al., 2011; Venkatakrishnan et al., 2013) require denoisers that have well defined priors and parameters. This means that the algorithms are not robust to any incorrect setup for a specific application. One of the causes of the strict requirements for the denoiser is that many denoisers are non-linear and do not have closed-form solutions. Therefore, many denoisers must be calculated iteratively, taking much more time and computational resources to calculate than a closed-form solution. In the classic ADMM algorithm as given in (Boyd et al., 2011), knowing the prior $g(x)$ is closed, proper, and convex is key to proving the convergence of the algorithm for a specific denoiser, D_σ, with a known prior $g(x)$. Hence, in the case where the corresponding prior $g(x)$ for D_σ is not known and the method for deriving $g(x)$ is also not known, there is no guarantee that we can obtain the relevant prior $g(x)$. In the case without the prior $g(x)$ the convergence of the ADMM for the specific D_σ is not provable which, if the algorithm does not converge, can lead to issues with stability and lack of precision in the results. This is a significant issue for applications that require reliability and high precision in the output to reduce errors when using the results. The

original PNP ADMM, as defined by (Venkatakrishnan et al., 2013), is unstable when not precisely tuned for a specific use case. The consequences for improperly tuned parameters for a specific application are the results of the ADMM algorithm in that application being unstable. As discussed here, previous ADMM algorithms can have problems with instability and computational resource usage when they are not tailored precisely for a specific application.

2.3. Applications

Some potential applications that are outlined in the source paper include image super-resolution and single photon imaging. The main objective function of the super-resolution problem is given below:

$$f(x) = ||SHx - y||^2 \tag{2}$$

There are two special cases where x can be solved for efficiently. Firstly, when $S = I$ which is the case for Non-blind Deblurring and when $H = I$ which is the case for for Interpolation. The ADMM algorithm, accompanied by an appropriate denoiser, can restore the entire images from the collected data points. Whether these data points are individual photons or larger images, the algorithm is applicable. An area where this could be useful is in damaged images, such as one with minor blemishes like scratches the algorithm can assist with inpainting the details of the original image. Within the medical field, the super resolution effects can help enhance and reconstruct high quality CT scans. Furthermore, denoising can be very effective in reducing noise seen in MRI scans creating high quality images to be used by surgeons.

3. Method

Generally, image restoration problems can be parameterized by

$$\hat{x} = \underset{x}{\operatorname{argmin}} \; -\log p(y \mid x) - \log p(x), \tag{3}$$

where $x \in \mathbb{R}^n$ is an image altered by a noisy model and $y \in \mathbb{R}^n$ is the unaltered original image. This maximum-a-posteriori (MAP) estimation can then be rewritten into the general unconstrained optimization problem given by

$$\hat{x} = \underset{x}{\operatorname{argmin}} \; f(x) + \lambda g(x) \tag{4}$$

the idea of ADMM is to convert this unconstrained optimization problem to the following constrained optimization problem:

$$(\hat{x}, \hat{v}) = \underset{x,v}{\operatorname{argmin}} \; f(x) + \lambda g(v), \text{ subject to } x = v \tag{5}$$

This constrained optimization problem can be broken down into the three key equations of ADMM using a variant of its augmented Lagrangian, which allow for convergence to an approximate solution by solving for x then v at each step. This version of the problem also allows for certain methods of quickly obtaining the solution non-iteratively.

3.1. Fixed Point Convergence

One of the primary contributions from this paper includes the proof that fixed point convergence can be achieved with the proper assumptions and bounds applied to the new PNP ADMM. The original paper (Chan et al., 2016) states, "Fixed point convergence guarantees that a nonlinear algorithm can enter into a steady state asymptotically". Put mathematically, there exists optimal terms (x^*, v^*, u^*) from the Lagrangian function that satisfy:

$$||x^{(k)} - x^*||_2 \to 0 \tag{6}$$
$$||v^{(k)} - v^*||_2 \to 0 \tag{7}$$
$$||u^{(k)} - u^*||_2 \to 0 \tag{8}$$

as $k \to \infty$. Under the assumptions that the negative log likelihood function, $f : [0,1]^n \to \mathbb{R}$, has bounded gradients and D_σ is a bounded denoiser, fixed point convergence can be achieved. This bounded denoiser must abide by the limit:

$$||D_\sigma(x) - x||^2/m \leq \sigma^2 C \tag{9}$$

In this definition, η and σ are parameters. C is some universal constant that is independent of η and σ. Additionally, bounded denoisers are expected to be asymptotically invariant. The bounded gradients of f means that there exists a value of $L < \infty$ for which the equation:

$$||\nabla f(x)||_2/\sqrt{n} \leq L \tag{10}$$

holds for any $x \in [0,1]^n$.

3.2. Adaptive Update Rule

Along with the addition of a bounded denoiser to achieve fixed point convergence, this paper proposes a method for a continuation scheme. This is implemented as an adaptive update rule during the calculation of the Lagrangian sub-problems. First, using a proposal from (Venkatakrishnan et al., 2013) the sub problem, (v^{k+1}) is altered to include the denoising algorithm,

$$v^{(k+1)} = D_{\sigma_k}(x^{(k+1)} - v^{(k+1)}) \tag{11}$$

with $\sigma_\kappa = \sqrt{\lambda/\rho_\kappa}$ as a weight parameter. Once the modification is done we can use the adaptive rule to solve for convergence. The update rule calculates the L2 Norm distance using each of the Lagrangian sub-problems, using the following equation,

$$\Delta_{k+1} = \frac{1}{\sqrt{n}}(||x^{(k+1)} - x^k||_2 + ||v^{(k+1)} - v^k||_2 + ||u^{(k+1)} - u^k||_2) \tag{12}$$

It is able to use the current and previous iterations of the sub-problems to identify a change in the distance to consider whether the update was successful or not. The criteria for a successful change is based on whether the change in the residue is larger than a set tolerance value, η. The advantage of this scheme is lies in the fact it is able to perform the ADMM algorithm quicker and in a more robust form. Additionally the inclusion of the plug and play, bounded denoiser, along with the adaptive updating scheme is vital for the fixed point convergence of the algorithm.

3.3. Fast Solver Implementation

In (Chan et al., 2016), the authors mainly discuss two fast implementation techniques, which are image super resolution and single photon imaging problems.

3.3.1. Image Super Resolution

The objective function for solving Image Super Resolution is defined as on equation (13), where $H \in \mathbb{R}^{n\times n}$ is a circulant matrix that representing the convolution for the anti-aliasing filter.

$$f(x) = ||SHx - y||^2 \tag{13}$$

Additionally, $S \in \mathbb{R}^{n\times n}$ is defined as a binary sampling matrix. We can simplify the objective function by substituting $G = SH$. If we plug equation (13) into equation (14), we can observe that our problem becomes as equation (15), and our solution would match equation (15) using pseudo-inverse. However, this is computationally expensive due to nature of inverse operation in matrix.

$$x^{(k+1)} = \arg\min_{x\in\mathbb{R}^n} \left(f(x) + \frac{\rho}{2}\|x - \tilde{x}^{(k)}\|^2 \right), \tag{14}$$

$$\hat{x} = \arg\min_{x\in\mathbb{R}^n} \|Gx - y\|^2 + \frac{\rho}{2}\|x - \hat{x}\|^2. \tag{15}$$

$$\hat{x} = (G^TG + \rho I)^{-1}(G^Ty + \rho\hat{x}). \tag{16}$$

We can avoid taking the inverse operation if our matrix is either diagonal or diagonalizable by Fourier transform. However, it is not true for the general case. There are two common ways to solve this task discussed in the literature. The first method is multi-variable splitting, which requires additional Lagrange multipliers and internal parameters. Therefore, it leads to slower convergence rate than the single variable optimization problem. The second method is inner conjugate gradient, which is computationally expensive. Therefore, authors proposed a polyphase decomposition to deal with the above issues. In order to use polyphase decomposition, there are two necessary conditions that need to be met. First, we must ensure that S is the standard K-fold downsampler. Second, H is a circular convolution. Then it is possible to derive a closed-form solution. The derivation of closed form solution begins with applying the Sherman-Morrison-Woodbury identity, which allows us to re-write the equation as

$$\tilde{x} = \rho^{-1}b - \rho^{-1}G^T(\rho I + GG^T)^{-1}Gb \tag{17}$$

The critical step is that we can leverage following equation

$$GG^T = SHH^TS^T \tag{18}$$

It is important to notice that we can obtain an upsampler filter and a downsampler sequence $S\hat{H}S^T$ by defining downsampler as $\hat{H} = HH^T$ and upsampler as S^T. We can obtain the downsampler by leveraging polyphase decomposition as shown in equation (19), where $\hat{H}_k(z^K)$ is the kth polyphase component of $\hat{H}(z)$. Therefore, we can simplify this operation SHH^TS^T by applying a finite impulse response filter $\hat{H}_0(z)$, which is 0th polyphase component of the filter HH^T. As a result, we can avoid inverse operation in (16) by using 0th polyphase component as shown on (20).

$$\hat{H}(z) = \sum_{k=0}^{K-1} z^{-k}\hat{H}_k(z^K) \tag{19}$$

$$x = \rho^{-1}b - \rho^{-1}G^T\mathcal{F}^{-1}\left\{\frac{\mathcal{F}(Gb)}{|\mathcal{F}(\hat{h}_0)|^2 + \rho}\right\} \tag{20}$$

3.3.2. Single Photon Image

The second fast implementation is using PNP ADMM for spatial oversampling device quanta imaging sensor (QIS). It is composed of collection of tiny detectors known as jots. The QIS uses multiple k jots to capture light corresponding to what would typically be one pixel in a standard sensor. For an image with n pixels, nK jots are utilized in total. We can further simplify the imaging model for the QIS that assumes a homogeneous distribution of light within each pixel. This model connects the underlying image x in $\mathbb{R}^n$ with the actual photon arrival rate at the jots s in $\mathbb{R}^{nK}$. As a result, the photon arrival rate becomes equation (21), where G is normalized by K diagonal matrix and α is sensor gain.

$$s = \alpha Gx \tag{21}$$

Given s, photon arriving at sensor can be illustrated as Poisson distribution with a rate given by s. We define Z_i as random variable of the number of photons at jot i, we obtain the Poison Distribution as in the below equation.

$$p(z_i) = \frac{s_i^{z_i}e^{-s_i}}{z_i!}, \quad i = 1, \ldots, nK. \tag{22}$$

Our QIS output, Y_i is a binary bit that uses threshold q, and when $q = 1$, the probability of observing $Y_i = y_i$ given s_i becomes as follows.

$$Y_i = \begin{cases} 1, & \text{if } Z_i \geq q, \\ 0, & \text{if } Z_i < q \end{cases} \tag{23}$$

$$p(y_i|s_i) = \begin{cases} e^{-s_i}, & \text{if } y_i = 0, \\ 1 - e^{-s_i}, & \text{if } y_i = 1. \end{cases} \tag{24}$$

By using defined conditional probability, we can estimate x from observed binary bits y with the recovery function

$$\begin{aligned} f(x) &\overset{\text{def}}{=} p(y|s) \\ &= \sum_{i=1}^{nK} -\log p(y_i|s_i) \\ &= \sum_{i=1}^{nK} -\log((1-y_i)e^{-s_i} + y_i(1-e^{-s_i})) \\ &= \sum_{j=1}^{n} -K_j^0 \log(e^{-\frac{\alpha x_j}{K}}) - K_j^1 \log(1 - e^{-\frac{\alpha x_j}{K}}) \end{aligned} \tag{25}$$

The K_j^1 is defined as the number of ones in the j pixel and

K_j^0 is the number of zeros in the j pixel. We can again plug our equation (25) to equation (14) to obtain our optimization problem.

$$\min_x \sum_{j=1}^{n} -K^0 \log\left(e^{-\frac{\alpha x_j}{K}}\right) - K^1 \log\left(1 - e^{-\frac{\alpha x_j}{K}}\right) + \frac{\rho}{2}(x_j - \tilde{x}_j)^2. \quad (26)$$

We can further simplify our optimization problem into one dimensional root finding problem by solving x_j independently.

$$Ke^{\frac{\alpha x_j}{K}}(\alpha + \rho(x_j - \tilde{x}_j)) = \alpha K^0 + \rho K(x_j - \tilde{x}_j), \quad (27)$$

4. Experiments

This section will outline our re-implementation[1] of Dr. Chan's PNP ADMM algorithm with simple examples of the usage. Then we will explore the application of PNP ADMM super-resolution and denoising in the area of traffic sign identification by a Machine Learning algorithm.

4.1. Example Use Cases

4.1.1. General Case

One of the primary applications of the PNP ADMM algorithm is image restoration. For this test implementation, an image was given into the system and Gaussian noise was added. The general PNP ADMM is solving MAP problem with splitting variable optimization with equations as shown below. The major advantage of using PNP ADMM over general ADMM is the freedom to choose the denoising algorithms.

$$x^{(k+1)} = \arg\min_{x \in R^n} \left(f(x) + \frac{\rho}{2}\|x - \tilde{x}^{(k)}\|^2\right), \quad (28)$$

$$v^{(k+1)} = D_{\sigma_k}(x^{(k+1)} + u^{(k)}), \quad \text{where } \sigma_k = \sqrt{\lambda/\rho_k}, \quad (29)$$

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \left(x^{(k+1)} - v^{(k+1)}\right) \quad (30)$$

4.1.2. Super-Resolution Case

The problem addressed by Super-Resolution is to increase the resolution of the image with minimal loss of detail. The objective function for solving Image Super Resolution is defined as on equation (13), where $H \in \mathbb{R}^{n \times n}$ is a circulant matrix that representing the convolution for the anti-aliasing filter. Additionally, $S \in \mathbb{R}^{n \times n}$ is defined as a binary sampling matrix. We can simplify the objective function by replacing with $G = SH$. If we plug equation (13) into equation (14), we can observe that our problem becomes as equation (15), and our solution would match equation (15) using pseudo-inverse. However, this is computationally expensive due to nature of inverse operation in matrix. As we have shown with equations (16) and (20) we can avoid having to compute the inverse operation, thus allowing for much faster execution.

[1] https://github.com/JStoschek/ECE50024-Team20-Final-Code

4.1.3. Results

Peak signal to noise ratio (PSNR) charts are used to gauge the performance of the algorithm. This chart compares the original image to the error introduced by the algorithm for either image reconstruction or super-resolution. Images with different resolutions were used during testing.

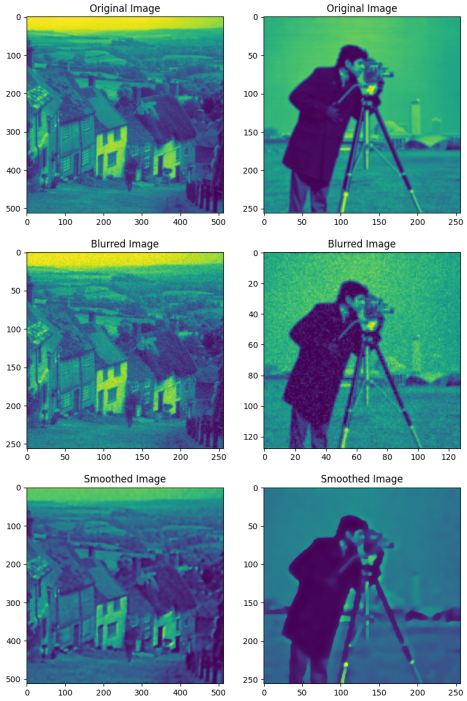

Figure 1. Deblurring results of our Plug and Play ADMM Algorithm

4.2. Traffic Sign Identification

The problem that we are addressing with our ADMM is the quality of the input to a decision making algorithm or neural network for autonomous systems. This has real world implications now with the expanding market for Autonomous Vehicles (AV) that rely on autonomous control systems that use video input. To this end we are using a dataset with images of traffic signs to explore the possible improvement from the higher quality image obtained by utilizing the ADMM as input to the decision making. We will be comparing the results of using our ADMM to a generic upscaling algorithm with the results of the classification after upscaling the images.

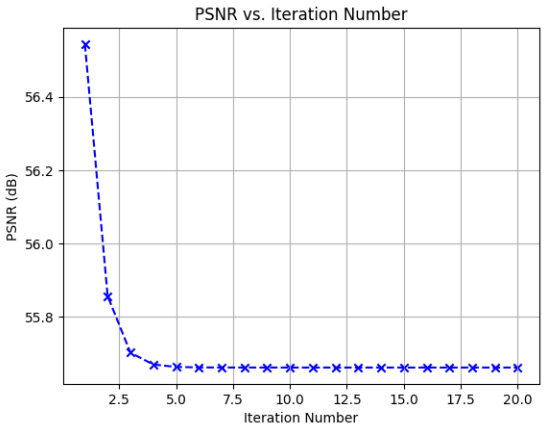

Figure 2. This is the PSNR(dB) plotted against the iteration number for the Hills.png image

4.2.1. Problem Description

We would like to apply our ADMM algorithm that has been designed for de-blurring and super resolution applications to road sign images. As AVs have become more popular, there has been a larger focus on their ability to notice and detect dangerous situations. We see instances of AVs causing fatal collisions because of inaccurately classifying the situation and therefore partaking in the incorrect action. For example, in 2016 a Tesla Model S in full self driving mode collided with a white tractor trailer while turning onto a highway, causing the death of the driver of the AV (Magnussen et al., 2020). The inability for the Tesla to correctly detect the trailer against the bright sky may have been the reason it did not apply the brakes. This shows that the detection systems of the AV are still affected by images that are not absolutely clear or contain noise. We also can see the use for improving traffic sign recognition, which has the same effect on AV safety. With our algorithm we would like to improve the images that the traffic sign recognition system uses in order for it to better classify scenarios and thus take the correct action.

The idea of using image super-resolution to detect road signs at a large distance has become important to how autonomous vehicles function. Similar networks, such as CNN based reconstructions (Wang et al., 2019) have been applied in the past. These are now being used for higher-fidelity object detection, as seen in (Wang et al., 2021).

4.2.2. Data

In order to obtain accurate prediction from deep learning models, it is important to prepare high quality dataset. Therefore, we decided to use the German Traffic Sign Benchmark dataset (Houben et al., 2013). It consists of 43 classes, and is from a single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011 (Stallkamp et al., 2012). The dataset has 3 distinct features, which are shape of sign, color of sign, and sign id following Ukrainian traffic rule. The images are taken from a 10h video taken with a camera installed in a car driving on various German roads (Stallkamp et al., 2012). There are more than 50,000 images of German road signs, so it would help our model learn with a sufficient amount of dataset.

4.2.3. Experiment

In order to test the efficacy of the PNP ADMM algorithm in our real-world application we propose an experiment to test the algorithm as a preliminary step to image classification. Our hypothesis is that the inclusion of the PNP ADMM algorithm will increase classification accuracy. The first step of our experiment is to augment the dataset by resize all the images to 64 by 64 pixels and applying a predetermined noising kernel to the image. This is crucial to PNP ADMM as it means we can use the best denoiser for the task. For the baseline experiment without PNP ADMM we will upsample the images using interpolation, so the images match the size of the output of PNP ADMM. We will then train a classifier and record the performance metrics (accuracy, precision, and recall) for comparison. For the second experiment we will use PNP ADMM to perform super-resolution and denoising and then train the same classifier. We will then compare the performance metrics of the two experiments. We expect to see a higher performance with the classifier that used PNP ADMM for upsampling compared to the classifier which used interpolation. If our results are statistically significant using a t-test, we know that the PNP ADMM algorithm has real world improvements and uses. This will most likely be due to PNP ADMM being configured for the ideal denoiser for the noise applied to the testing images.

In an experiment to test this hypothesis, we used the the German Traffic Sign Benchmark dataset (Houben et al., 2013) we performed image prepossessing to compare our model against. Most of the images within the dataset had a varying image size so the first step we took was to resize all of the images to a standard 32 by 32 pixels. Making the images a standard size will not only make the images easier to upload and work with but there are also training advantages when all of the images are standardized.

Gaussian noise with zero mean was added to the images to better exemplify the use of the algorithm. We needed a way to understand how the processed images cause differences in the classifying performance. If we are able to test the performance of our classifying network against two different datasets. The noised images would then be up-sampled by either interpolation for the controll experiemnt, or our PNP ADMM algorithm for the test experiment.

4.3. Image Super Resolution in Classification

4.3.1. ADMM Modifications

Once the input images were prepared by adding noise and resizing them, there was still a few steps left before the PNP ADMM algorithm can successfully make an altered image. The PNP ADMM implementation only works on 1 channel gray scale images so it had to be updated to work on the color images of the dataset. This was done by splitting each image into its corresponding RGB channels and running them individually through our PNP ADMM implementation. Finally we combine each channel and normalize to obtain the final color image for out testing dataset.

4.3.2. Classifier Setup

For get our evaluation metric for the experiment, we used a Pytorch implementation of Faster R-CNN (Girshick, 2015) with backbone network of Mobile Net (Howard et al., 2017). The Faster R-CNN is an object detection model that can effectively capture the object in the image. The major component of Faster R-CNN is composed of a backbone network, which extracts the features from the image, a regional proposal network, which generates the anchor boxes by efficient sliding window approach, and a fully connected layer, which outputs both regression deltas and classification scores as shown on Figure 3 (Girshick, 2015). We modified the fully connected layer at the end of F-RCNN to predict the 43 classes from our dataset.

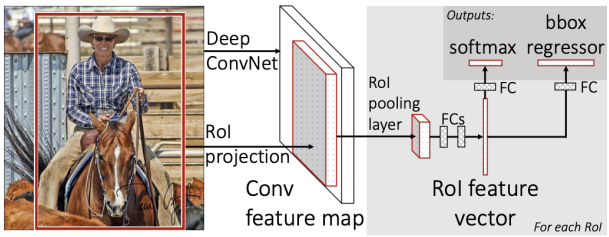

Figure 3. Architecture of F-RCNN (courtesy of (Girshick, 2015))

The base images were created as lower resolution images at 32 by 32 pixels with Gaussian noise. After using our PNP ADMM, the images were 64 by 64 pixels and denoised. The classifier network was initialized from a model pretrained on the COCO dataset and fine-tuned with our dataset for 150 epochs of training.

4.3.3. Performance

An example can be seen below showing an original stop sign image from the dataset, then go through the processing steps before feeding into the classifier.

Parameter	Value
SGD Learning Rate	5e-5
Momentum	0.9
Weight Decay	1e-3
Batch Size	34

Table 1. Model and Optimizer parameters for classifier training

Figure 4. Unmodified stop sign (left) stop sign after noise and upsampling (middle) stop sign after noise + PNP ADMM (right)

The effects of increasing the resolution can be seen below.

Condition	Accuracy (%)
Before Super Resolution	43
After Super Resolution	55

Table 2. Comparison of accuracy before and after PNP ADMM super resolution

As expected, applying super resolution to the dataset images to double their size increased the performance. Using the Faster R-CNN on these two datasets yielded 12% increase in performance when the super resolution was utilized before passing the data into the Faster R-CNN.

5. Conclusions

Implementing the PNP ADMM super resolution yielded the desired result and increased the performance of the F-RCNN algorithm when being applied to images of road signs. This is in line with results seen in (Wang et al., 2019), which also applied super resolution to small images of road signs to increase the detection rate compared to searching the original image.

This result encourages the use of the PNP ADMM in a situation such as automated driving. The ability to detect road signs from a far distance, when the resolution of the road sign is particularly small, provides opportunities for better path planning and a smoother driving experience. Other implementations and experimentation could be applied further than just road signs, and could improve the detection range of the vehicle as a whole.

6. Contributions

Kyung Min Ko: Kyung Min worked on implementing PNP ADMM algorithm with focusing on main loop of the algorithm. Modified the F-RCNN to predict the class from road sign dataset. Worked on writing mathematical derivation of the algorithm and classification paragraph in the experiment section.

Kunal Mamtani: Was in charge of applying image reprocessing to the dataset, including resizing and adding noise. Implemented the Toy implementation to find images processed by the PNP ADMM. Created graphs to show noise within the image plotted against training iteration. Worked on creating Initial Implementation Results section, Adaptive Update rule, Problem Description section and Experimental Setup descriptions.

Jonathan Stoschek: Edited the PNP_ADMM.py script to work with color images and constructed testing and control datasets for real world evaluation. Documentation and comments for rcnn.py script as well as the PNP_ADMM.py script. Developed experimental procedure and wrote corresponding section in checkpoint 4 and first part of section 4.2.3 in the final report as well. Additionally, helped write first part of Section 2 Problem Statement, Sections 2.3, 3.3.1, 4.1.2, 4.3.1, added figures. Finally, responsible for organizing files and managing GitHub.

Heron Teegarden: Worked on the initial implementation of the low level PNP ADMM functions and dataset research for applicable datasets in our real world problem. Wrote Overall Introduction, Why PNP ADMM, Traffic Sign Identification, and assisted in writing and editing many of the other sections.

Alec Vucsko: Wrote sections 3.1, 4.1.1, 4.3.3, 5. Assisted writing other sections in the paper such as 2, 3, 4.2.3, and edited others. Conducted road sign super resolution research and narrowed down relevant papers. Built early versions of ADMM afun and gfun in python.

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Chan, S. H., Wang, X., and Elgendy, O. A. Plug-and-play ADMM for image restoration: Fixed point convergence and applications. *CoRR*, abs/1605.01710, 2016. URL http://arxiv.org/abs/1605.01710.

Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013. URL https://api.semanticscholar.org/CorpusID:700906.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Magnussen, A., Le, N., Hu, L., and Wong, E. W. A survey of the inadequacies in traffic sign recognition systems for autonomous vehicles. *International Journal of Performability Engineering*, 16, 2020.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32: 323–332, 2012.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE global conference on signal and information processing*, pp. 945–948. IEEE, 2013.

Wang, F., Shi, J., Tang, X., Guo, J., Liang, P., and Feng, Y. Cnn-based super-resolution reconstruction for traffic sign detection. *IEEE Symposium Series on Computational Intelligence*, 2019.

Wang, Y., Yang, J., Yin, W., and Zhang, Y. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3): 248–272, 2008.

Wang, Z.-Z., Xie, K., Zhang, X.-Y., Chen, H.-Q., Wen, C., and He, J.-B. Small-object detection based on yolo and dense block via image super-resolution. *IEEE Access*, 9, 2021.