Image classification and training with severe data loss

Dillon Marquard, Kyle Wright, Roummel Marcia

Dillon Marquard, Kyle Wright, Roummel Marcia, "Image classification and training with severe data loss," Proc. SPIE 12227, Applications of Machine Learning 2022, 122270T (3 October 2022); doi: 10.1117/12.2633172

Event: SPIE Optical Engineering + Applications, 2022, San Diego, California, United States

Image classification and training with severe data loss

Dillon Marquard[a], Kyle Wright[a], and Roummel Marcia[a]

[a]University of California Merced, 5200 Lake Rd, Merced, CA

ABSTRACT

Image classification forms an important class of problems in machine learning and is widely used in many real-world applications, such as medicine, ecology, astronomy, and defense. Convolutional neural networks (CNNs) are machine learning techniques designed for inputs with grid structures, e.g., images, whose features are spatially correlated. As such, CNNs have been demonstrated to be highly effective approaches for many image classification problems and have consistently outperformed other approaches in many image classification and object detection competitions. A particular challenge involved in using machine learning for classifying images is measurement data loss in the form of missing pixels, which occurs in settings where scene occlusions are present or where the photodetectors in the imaging system are partially damaged. In such cases, the performance of CNN models tends to deteriorate or becomes unreliable even when the perturbations to the input image are small. In this work, we investigate techniques for improving the performance of CNN models for image classification with missing data. In particular, we explore training on a variety of data alterations that mimic data loss for producing more robust classifiers. By optimizing the categorical cross-entropy loss function, we demonstrate through numerical experiments on the MNIST dataset that training with these synthetic alterations can enhance the classification accuracy of our CNN models.

Keywords: Convolutional Image Classification, Machine Learning, Neural Network

1. INTRODUCTION

Image classification is a task that involves categorizing and labeling image data, and it forms an important class of problems in machine learning that is widely used in a variety of real-world applications, such as medicine,[1–3] ecology,[4–7] astronomy,[8–10] and defense.[11–13] Often, these images are obtained in less than ideal environments. For example, medical, astronomical, and night-vision images are typically captured in low-photon count settings, causing the measurements to be corrupted by noise artifacts.[14,15] Atmospheric turbulence often introduce blur in remote sensing[12] while moving objects within scenes produce streaks and motion blur.[16] In this work, we explore the performance of a specific type of machine learning technique for image classification on a different type of image corruption, namely the loss of pixel intensity information. Such data loss can occur in the presence of scene occlusions[17] or of defective pixels in a sensor array.[18] We consider convolutional neural networks (CNNs),[19] which are machine learning techniques designed for inputs with grid structures, e.g., images, whose features are spatially correlated. As such, CNNs have been demonstrated to be highly effective approaches for many image classification problems and have consistently outperformed other approaches in many image classification and object detection competitions.

The paper is organized as follows. We review in detail the CNN model in Sec. 2. In Sec. 3, we describe the datasets that were used in our numerical experiments, which we present in Sec. 4. We conclude with a discussion of the results in Sec. 5.

Further author information:
Dillon Marquard: Email: dmarquard@ucmerced.edu
Kyle Wright: Email: kwright11@ucmerced.edu
Roummel F. Marcia: Email: rmarcia@ucmerced.edu

Applications of Machine Learning 2022, edited by Michael E. Zelinski,
Tarek M. Taha, Jonathan Howe, Proc. of SPIE Vol. 12227, 122270T
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2633172

2. CONVOLUTIONAL NEURAL NETWORKS

Neural networks are models with a specific structure that are frequently used to approximate unknown functions. In the case of feed-forward neural networks, this structure consists of an input layer, some number of hidden layers, and a final output layer. In addition, these layers are connected using parameters, referred to as *weights*, and nonlinear activation functions. While a neural network containing weights comprised of random values is unlikely to accurately estimate the desired function, a network can improve its approximation through the process of updating these weights, referred to as *training*.

CNNs are a class of feed-forward neural networks frequently seeing use in image classification where the weights trained are inside convolutional filters. Image data can be thought of as a 2D array that represents the intensity of light incident on a set of pixels. These filters, at a high level, extract features from the image data. After training, these features can then be used to predict the class of an image by associating the output layer of the model with a probability it resides in each class. The classification is assigned to the class with the highest observed probability.

The baseline CNN we chose for our model has 4 convolutional layers and then 4 fully-connected linear layers and a final output layer passed through a softmax function. Each layer uses a ReLU activation function. The step size of our filter is 1, and the filter sizes are depicted in Fig. 1. We chose a simple convolutional model to reduce training time, so that we could explore a larger variety of training sets. In particular, we wanted to understand if training the model on certain noise and corruption patterns made models robust to other types of noise and corruption patterns. We were somewhat limited computationally, so our investigation did not cover the prevalence of model size. Additionally, we chose a smaller model size because we wanted to avoid overfitting, which tends to occur with larger models that can memorize the training set.

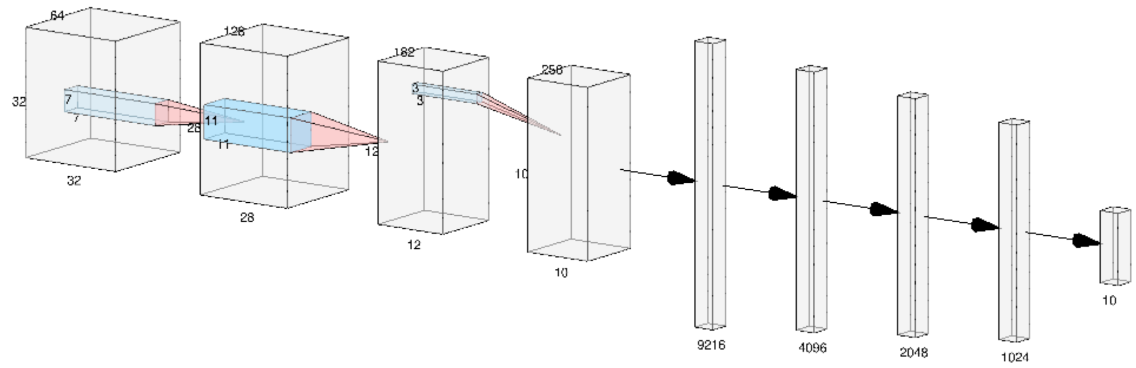
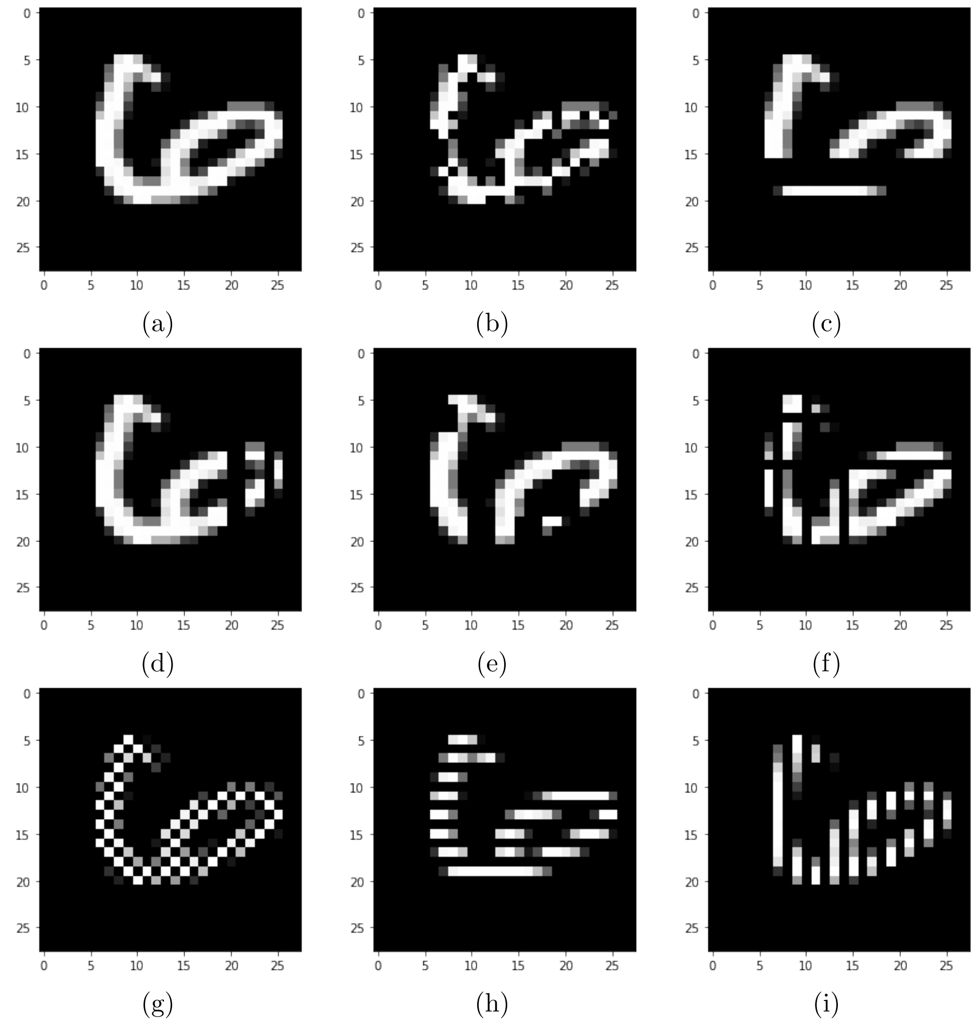


Figure 1. Illustration of the CNN structure used for baseline model.

If a CNN is classifying accurately, then the class predictions of the network will match the true class label. Consider the case of optimizing the parameters for N training images and M classes where index i corresponds to the image number and index j corresponds to the class. Let $\mathbf{y}$ be the vector of true labels (assigned value 1 for the true class and 0 for all other classes), $P(\mathbf{x}_i;\theta)$ be the network's prediction (a vector containing the probability the image belongs to each class) given the input image $\mathbf{x}_i$ and neural network weights θ. Then, we can formulate the following minimization problem using the cross-entropy loss function:[20]

$$\underset{\theta}{\text{minimize}}\ f(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}[\mathbf{y}_i]_j \log([P(\mathbf{x}_i;\theta)]_j).$$

Naturally, the loss function–and thus the optimal θ–are dependent upon the training data. Consequently, when the network is trained on images that are sufficiently different from the testing images, it can lead to inaccurate classification. In the next section, we describe the various data sets that we used in our numerical experiments.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 2. Patterns of missing pixels in the dataset. (a) Ground truth. (b) Salt and pepper. (c) Random rows. (d) Random columns. (e) Random block 3 × 3. (f) Random rows and columns. (g) Checkerboard. (h) Every other row. (i) Every other column.

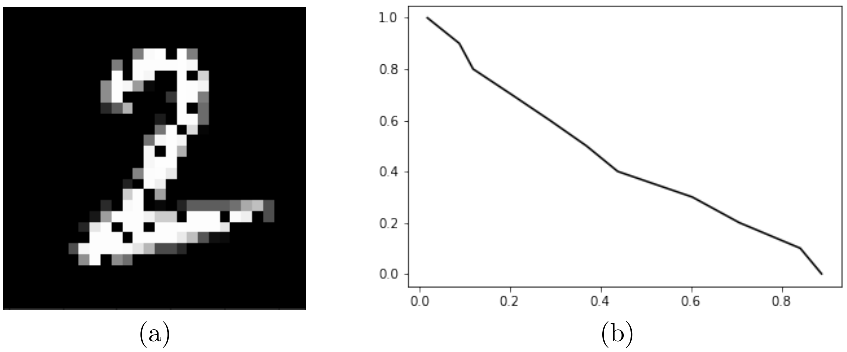
3. DATA SETS

The dataset we used for our investigation is the MNIST dataset.[21] This dataset consists of 70,000 handwritten digits roughly evenly distributed amongst the digits 0 through 9. Each image is represented as a 28×28 array of values ranging from 0 to 1. This dataset is arguably one of the standards for testing image classification methods. Here, we explore three different types of pixel data loss, which will be used in our numerical experiments:

I. **Removal:** A subset of the pixel locations is chosen and the corresponding pixel values are set to 0, indicating missing information at these locations.

II. **Replacement:** A subset of the pixel locations is chosen and the corresponding pixel values are set to random values drawn from a uniform distribution between 0 and 1, indicating incorrect measurements at these locations.

III. **Removal/Replacement Combination(Composite):** The first method removes the pixel data, and the second method replaces the pixel data with random values between 0 and 1.



(a)

(b)

Figure 3. (a) Example of an image where 20% of the pixels are removed. (b) Performance curve of a model trained on the original MNIST data (without missing pixels) for classification accuracy as a function of missing pixel percentage .

The subset of pixel locations falls under two main categories: (A) random and (B) fixed. For the random subset category, we consider the following patterns:

i. **Salt and pepper:** Pixel coordinates are uniformly sampled.

ii. **Row:** Row indices are uniformly sampled.

iii. **Columns:** Column indices are uniformly sampled.

iv. **Block** 3×3**:** Center pixel coordinates of the 3×3 blocks are uniformly sampled.

v. **Rows and columns:** Row and column indices are uniformly sampled.

We note that the subsets are chosen without redundancy, meaning the same row or column can be chosen twice in the same subset. For the fixed subset, we consider the following patterns:

i. **Checkerboard:** Every other pixel is removed with an alternating pattern to create a checkerboard pattern.

ii. **Horizontal stripes:** Every other row is removed.

iii. **Vertical stripes:** Every other column is removed.

We illustrate these patterns in Fig. 2.

In our numerical experiments in Sec. 4, we explored various noise patterns for creating the training data. After training a network using a particular combination of simulated noise type and pattern(s), we can gauge the robustness of our network by its ability to accurately classify images from sets using other combinations of simulated noise types and patterns. For example, we tested the performance of a model trained on data with half of the rows randomly removed on data with half of the columns randomly removed.

For each experiment, a specific pixel loss pattern and type are chosen. When the models are trained, at each epoch, a different dataset with the same loss pattern and type is used. For randomly generated patterns, different random choices are used. In order to validate our results we split the dataset into a training and testing batch. The MNIST dataset is partitioned with 60,000 training images, and 10,000 testing images. The dataset is randomly shuffled before batch sampling.

For models that use more than one pixel loss pattern, we train by equally splitting the training set between each pattern. It is important to note that the set was randomly shuffled after splitting between the pixel loss patterns. This is different from our composite models, which use a combination of patterns when selecting pixel loss type.

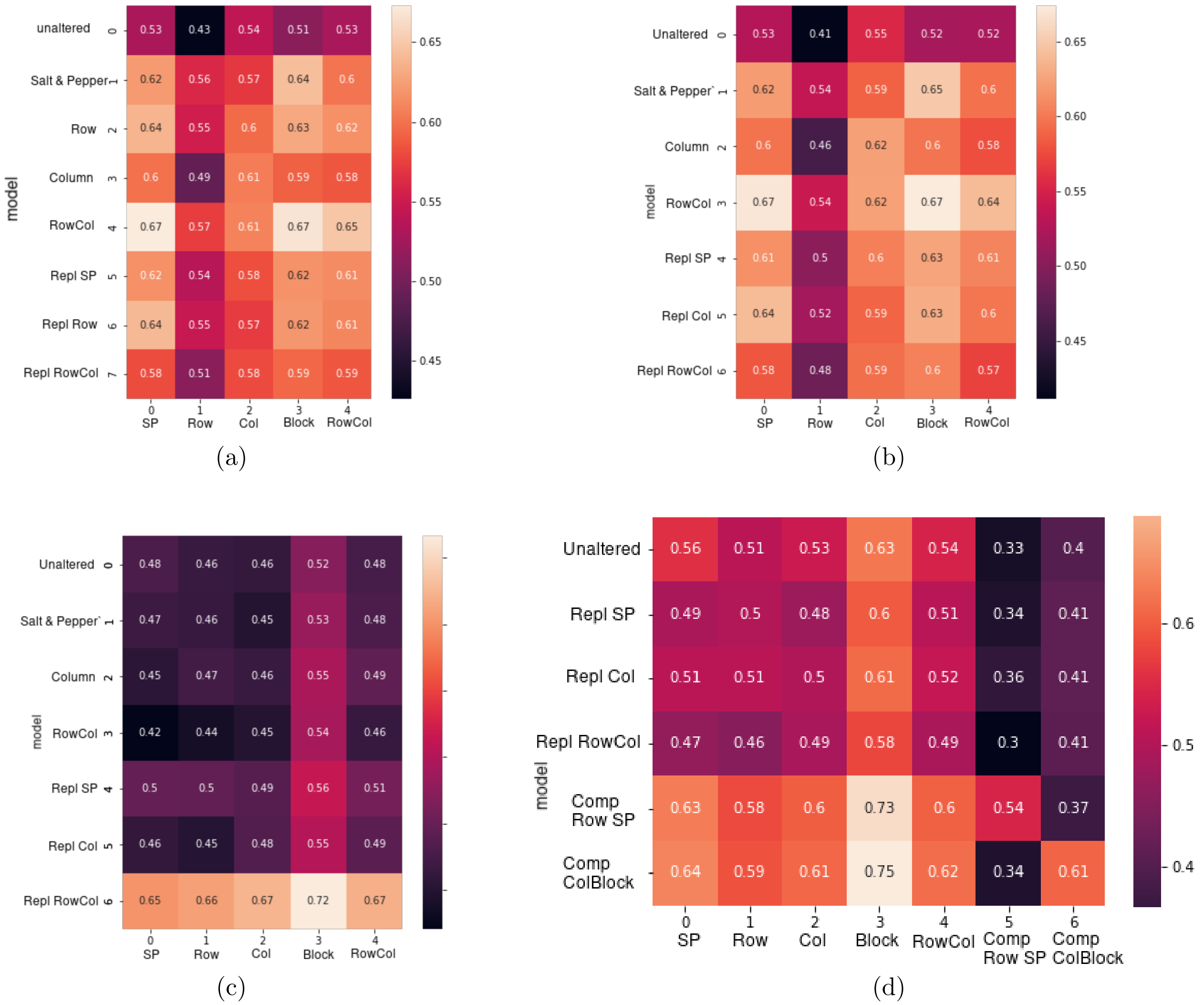
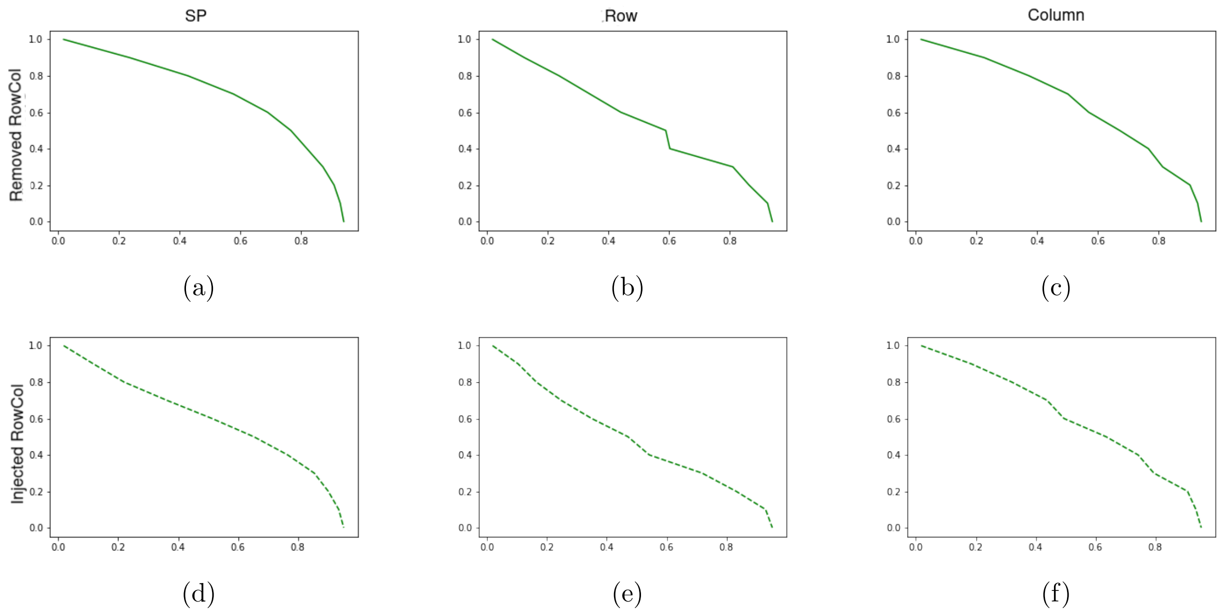


Figure 4. Area under the curve (AUC) scores for various training and testing data loss types and patterns. (a) Both training and testing sets have missing pixels i.e., both are of the removal type. (b) The training set is of the removal type, and the testing set is of the replacement type, i.e., some pixels have been replaced by values randomly drawn from a uniform distribution. (c) The training set is of the replacement type, and the testing set is of the removal type. (d) Training set is of replacement and removal/replacement combination types, and tested on combination type.

4. NUMERICAL EXPERIMENTS

We tested our baseline CNN model on the unmodified MNIST dataset and found it to perform with a high accuracy of 98%. We found that adding a small amount of noise to the image data had a large negative impact on the model's performance. Often, these models are deployed with some form of preprocessing to the input image data. These preprocessing steps may only fix specific problems with input data and do not generalize well to a robust set of adverse image pixel loss patterns. There is an ever expansive set of loss patterns that can be applied to an image, so we seek a more generally applicable solution to image noise and corruption. We sought an alternative solution to preprocessing the image data by incorporating various loss patterns into the training data. This results in a more robust model that renders a large variety of preprocessing techniques to filter the data less necessary. In these numerical experiments, we explore the capacity of using training data with various



(a) (b) (c)

(d) (e) (f)

Figure 5. Classification accuracy on a testing set of *removal* type where the training set is of the removal type ((a)-(c)) and of the replacement type ((d)-(f)). The training set has combined row and column missing pixel pattern, and the testing set uses salt and pepper ((a) and (d)), row ((b) and (e)), and column ((c) and (f)) pixel loss patterns.

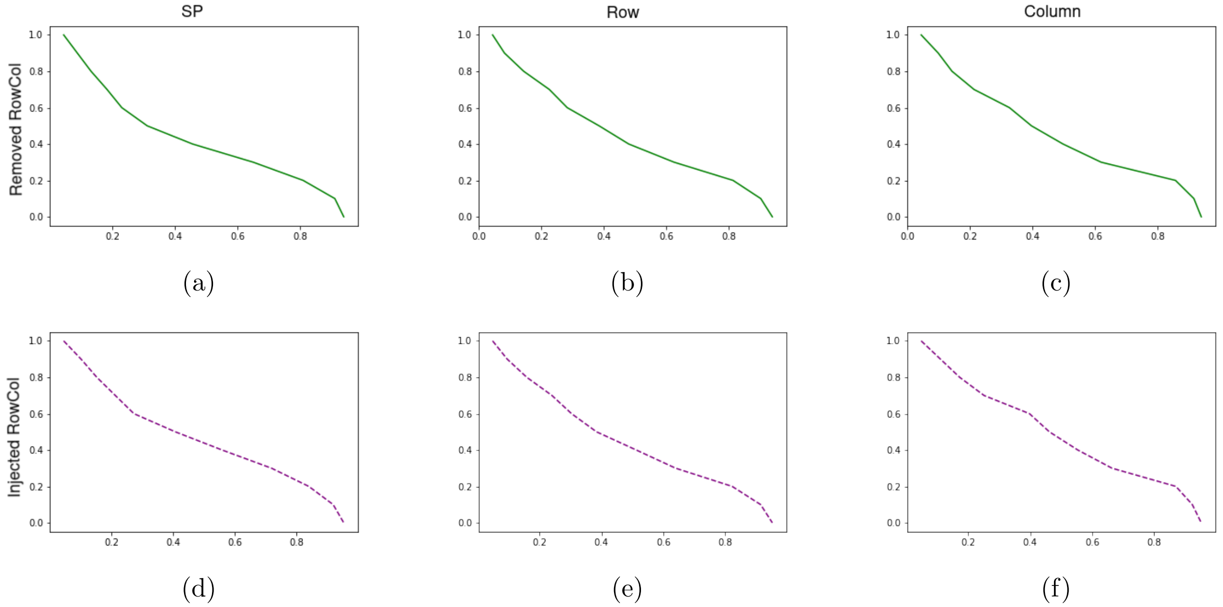
loss patterns to develop CNN models more robust to image pixel loss and corruption.

Fig. 3 illustrates an example of the correlation between pixel loss and model prediction accuracy. With a pixel loss of 20% randomly chosen (see Fig. 3(a)), the baseline models shows an approximately a 20% decrease in accuracy. In addition, we can observe that the number still appear clearly identifiable even with 20% of the pixels removed. This indicates that the decrease in performance could potentially be improved upon given the information remaining in the image. Moreover, we expected our baseline model to be more robust given the large set of training data. However, the model demonstrates a rapid decrease in accuracy, even when minimal alterations made to the testing image.

We examined the performance of the model for classification accuracy where the training and testing sets do not necessarily have the same pixel data loss type. We focus our analysis on the accuracy and precision of the models. Accuracy gives us a more general performance metric and precision has a greater fidelity for individual class performance.

We quantify this accuracy by computing the area under the curve (AUC) of the performance curves similar to Fig. 3(b) for these different data loss types. The AUC values range between 0 and 1 with a higher value indicating a better performance. In Fig. 4(a), we present the AUC for the various missing pixel patterns where the both the training and testing sets have missing pixels, i.e., both are of the removal type. Fig. 4(b) corresponds to the case where the training set is of the removal type, and the testing set is of the replacement type, i.e., some pixels have been replaced by values randomly drawn from a uniform distribution. Fig. 4(c) corresponds to the case where the training set is of the replacement type, and the testing set is of the removal type. Finally, Fig. 4(d) corresponds to the case where the training set is of replacement and removal/replacement combination types, and tested on combination type.

Naturally, we noticed that models trained on a given pixel loss pattern performed noticeably above the baseline model when classifying test images using that same loss pattern. However, the models trained on data with missing pixels also performed equally well with the ground truth data set on the ground truth data set (images without missing pixels). This suggests that the training on data with missing pixels did not significantly decrease the models' accuracy for classifying the ground truth images.



(a) (b) (c)

(d) (e) (f)

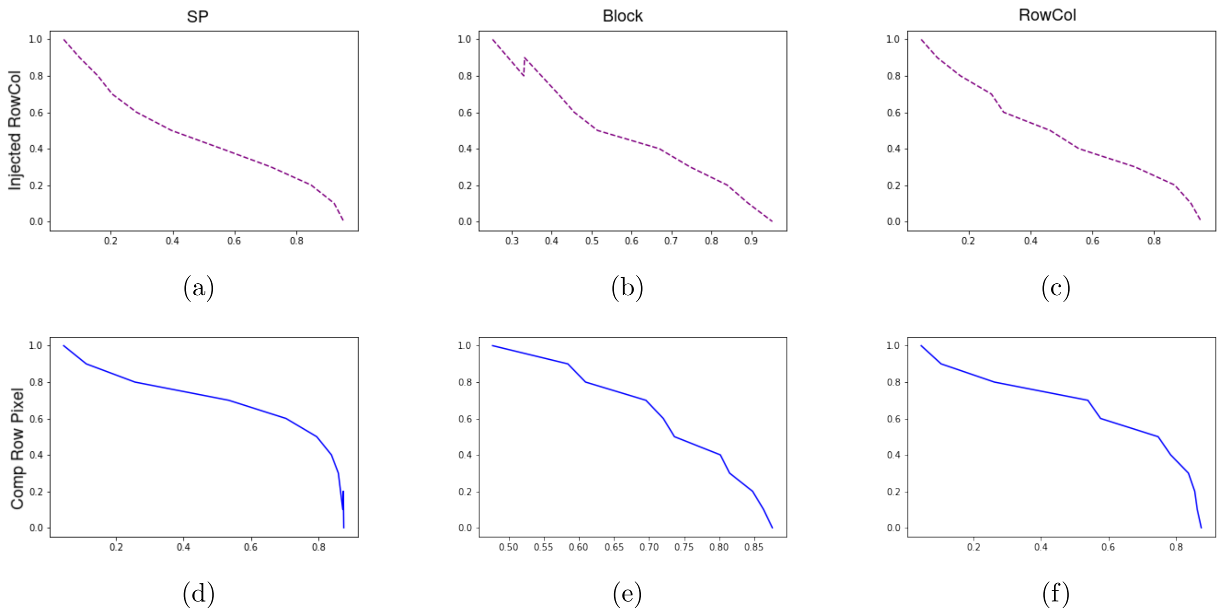
Figure 6. Classification accuracy on a testing set of *replacement* type where the training set is of the removal type ((a)-(c)) and of the replacement type ((d)-(f)). The training set has combined row and column missing pixel pattern, and the testing set uses salt and pepper ((a) and (d)), row ((b) and (e)), and column ((c) and (f)) pixel loss patterns.

We found that data sets with fixed patterns (checkerboard, horizontal stripes, and vertical stripes) did not produce our goal of robust models capable of accurately classifying images of a different pixel loss pattern. On the other hand, models incorporating randomness in the training data loss pattern greatly increased their performance in classifying images using other loss patterns. Models trained with a component of randomness were far more robust to a variety of loss patterns. We found that the models with fixed patterns performed far worse, and did not generalize well over the set of pixel loss patterns.

In addition, we found that models trained on data sets of the *replacement* type were more robust than models trained on those of the *removal* type. It should be noted that some models trained using data sets of removal type were robust with respect to various pixel loss patterns for testing sets of removal types. However, they were unable to accurately classify test images of *replacement* type. On the other hand, some models trained on data sets of *replacement* type were shown to classify accurately on both types of pixel data loss . This would imply the models are more robust when they have been trained using the replacement type rather than the removal type. We chose to further explore the ability to generalize over an even more robust set by increasing the number of pixel loss patterns present in the training data.

Fig. 5 illustrates the model's performance where the training data is a mixture of two pixel loss patterns (row and column) of the removal type (Fig. 5(a)-(c)) and of the replacement type (Fig. 5(d)-(f)). Here the testing sets are of removal type with salt and pepper (Fig. 5(a) and (d)), row (Fig. 5(b) and (e)), and column (Fig. 5(c) and (f)) pixel loss patterns. We used the combined row and column pattern because the corresponding results compared very well to the other patterns. Fig. 6 illustrates a similar set up with the difference that the testing sets are of the replacement type, where there is a significant decrease in performance. The AUC of the combined row and column pattern averaged 67% compared to the model trained on the ground truth (original data) with a maximum AUC of 52% (see Fig. 4). This prompted us to examine other combinations of these methods. In particular, we combined replacement and removal types to form the basis of our composite models.

We found that the composite model generalized the problem far better than any other model. We found this model to be particularly robust because even with 70% of the pixels missing the model maintained a high degree of precision for each class. We can see in Fig. 7, the composite model performs well across a majority of the pixel loss patterns. Moreover, the model maintains that degree of performance across increasingly large amounts



(a) (b) (c)

(d) (e) (f)

Figure 7. Classification accuracy on a testing set of *replacement* type where the training set is of the replacement type ((a)-(c)) and of the removal/replacement combination type ((d)-(f)). The training set has combined row and column missing pixel pattern, and the testing set uses salt and pepper ((a) and (d)), row ((b) and (e)), and column ((c) and (f)) pixel loss patterns. Comparison between replacement models(a) and composite models(b) on the SP, Block and RowCol replacement methods.

of loss patterns. This consistency is more noticeable when we compare it to non-composite models that seem to preferentially improve the precision of a small subset of the digits. We can see from Fig. 4 that the composite models have an AUC ≈ 0.1 above the models trained with replacement. Additionally, the composite models perform better than the baseline model, which indicates a particularly well rounded and robust model.

One limitation we found with composite models is that they do not perform as well when tested on data sets with other composite loss patterns. We can see this by comparing the AUC between the two composite models in Fig. 4. We can observe that composite models maintain high AUC scores when applied to non-composite loss patterns. In addition, we found that composite models outperform non-composite models when testing on other loss patterns. However, as the testing set incorporates increasing complexity in the loss patterns, even the composite models decrease in efficacy. This tells us that while we can increase the complexity of the image loss patterns, there is still a limit to the extent that combining a fixed number of different loss patterns will help improve performance.

5. DISCUSSION

The basis of our experiments was to quantify the robustness of our models. We find that robustness is a measure of the models capacity to maintain a high performance across increasing levels of pixel loss on variety of different loss patterns. We tested a limited set of loss patterns, so future studies should test a wider variety of patterns and generalized forms of pixel loss. Of the set we tested we found certain training methods to develop far more robust models than others.

In particular, composite data sets of *replacement* type and loss patterns *column* and *block* performed particularly well. We found that building more complex loss patterns types improved performance, but with diminishing returns as each form of pixel loss is layered on. We only explored two composite models, but future work could further explore different proportions of pixel loss and combinations. In addition, would also be useful to explore methods of determining what combinations of pixel loss patterns are most likely to improve model robustness for a type of dataset, as well as whether a pattern of combinations becomes apparent when implementing the method on various datasets.

The best training methods we produced included the following design patterns. Most notably, randomness greatly improved the performance of our models. Static methods leave many weights randomized and can have unwanted effects on the output layer. In addition, the models trained with data sets of the *replacement* type outperformed those of the *removal* type, but their combination, as we can see with composite models, is even better. In summary, it appears that the more diversity is incorporated into training a model, the more robust the model becomes.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation Grant DMS 1840265.

REFERENCES

[1] Erickson, B. J., Korfiatis, P., Akkus, Z., and Kline, T. L., "Machine learning for medical imaging," *Radiographics* **37**(2), 505 (2017).

[2] Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., and Chen, M., "Medical image classification with convolutional neural network," in [*2014 13th International Conference on Control Automation Robotics & Vision*], 844–848, IEEE (2014).

[3] Spanhol, F. A., Oliveira, L. S., Petitjean, C., and Heutte, L., "A dataset for breast cancer histopathological image classification," *IEEE Transactions on Biomedical Engineering* **63**(7), 1455–1462 (2015).

[4] Recknagel, F., "Applications of machine learning to ecological modelling," *Ecological modelling* **146**(1-3), 303–310 (2001).

[5] Tabak, M. A., Norouzzadeh, M. S., Wolfson, D. W., Sweeney, S. J., VerCauteren, K. C., Snow, N. P., Halseth, J. M., Di Salvo, P. A., Lewis, J. S., White, M. D., et al., "Machine learning to classify animal species in camera trap images: Applications in ecology," *Methods in Ecology and Evolution* **10**(4), 585–590 (2019).

[6] Wäldchen, J. and Mäder, P., "Machine learning for image based species identification," *Methods in Ecology and Evolution* **9**(11), 2216–2225 (2018).

[7] Christin, S., Hervet, É., and Lecomte, N., "Applications for deep learning in ecology," *Methods in Ecology and Evolution* **10**(10), 1632–1644 (2019).

[8] Ball, N. M. and Brunner, R. J., "Data mining and machine learning in astronomy," *International Journal of Modern Physics D* **19**(07), 1049–1106 (2010).

[9] Kremer, J., Stensbo-Smidt, K., Gieseke, F., Pedersen, K. S., and Igel, C., "Big universe, big data: machine learning and image analysis for astronomy," *IEEE Intelligent Systems* **32**(2), 16–22 (2017).

[10] De La Calleja, J. and Fuentes, O., "Machine learning and image analysis for morphological galaxy classification," *Monthly Notices of the Royal Astronomical Society* **349**(1), 87–93 (2004).

[11] Zhou, Y., Wang, H., Xu, F., and Jin, Y.-Q., "Polarimetric sar image classification using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters* **13**(12), 1935–1939 (2016).

[12] Zhu, P., Isaacs, J., Fu, B., and Ferrari, S., "Deep learning feature extraction for target recognition and classification in underwater sonar images," in [*2017 IEEE 56th Annual Conference on Decision and Control*], 2724–2731, IEEE (2017).

[13] Zhan, Y., Hu, D., Wang, Y., and Yu, X., "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters* **15**(2), 212–216 (2017).

[14] Yu, Z., Leng, S., Kappler, S., Hahn, K., Li, Z., Halaweish, A. F., Henning, A., and McCollough, C. H., "Noise performance of low-dose ct: comparison between an energy integrating detector and a photon counting detector using a whole-body research photon counting ct scanner," *Journal of Medical Imaging* **3**(4), 043503 (2016).

[15] Jerram, P., Pool, P. J., Bell, R., Burt, D. J., Bowring, S., Spencer, S., Hazelwood, M., Moody, I., Catlett, N., and Heyes, P. S., "The llccd: low-light imaging without the need for an intensifier," in [*Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II*], **4306**, 178–186, SPIE (2001).

[16] Dai, S. and Wu, Y., "Motion from blur," in [*2008 IEEE Conference on Computer Vision and Pattern Recognition*], 1–8, IEEE (2008).

[17] Shin, D.-H., Lee, B.-G., and Lee, J.-J., "Occlusion removal method of partially occluded 3d object using sub-image block matching in computational integral imaging," *Optics Express* **16**(21), 16294–16304 (2008).

[18] Ghosh, S., Marshall, I., and Freitas, A., "Autonomously detecting the defective pixels in an imaging sensor array using a robust statistical technique," in [*Image Quality and System Performance V*], **6808**, 378–389, SPIE (2008).

[19] Fukushima, K. and Miyake, S., "Neocognitron: Self-organizing network capable of position-invariant recognition of patterns," in [*Proc. 5th Int. Conf. Pattern Recognition*], **1**, 459–461 (1980).

[20] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H., [*The elements of statistical learning: data mining, inference, and prediction*], vol. 2, Springer (2009).

[21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), 2278–2324 (1998).