The Effect of Noise on Image Classification

Joseph Acernese[1]

[1]*School of Computer Science, University of Guelph*
50 Stone Rd E, Guelph, ON N1G 2W1
1

jacernes@uoguelph.ca

Abstract— **Image classification is a very popular topic within the realm of machine learning. However, most research in this field uses clean data with no noise, while real-world data is often affected by common errors and noise. Additionally, a vast majority of this research does not consider classic machine learning algorithms. This study analyzes the effect of visual noise on the performance of Support Vector Machines (SVM), K-Nearest Neighbours (KNN), Naive Bayes, and Random Forest. We conduct experiments on each classifier that considers various types of noise at varying intensities. Individual analysis of each classifier and comparative analysis of all classifiers are discussed in detail. Results suggest SVM and KNN are not fit to handle noisy images, while Naive Bayes and Random Forest are robust against noise. Naive Bayes presents with a lower overall accuracy but remains consistent, while Random Forest has a higher overall accuracy but is more variable across varying levels of noise.**

Keywords— **Image classification, visual noise, robustness, SVM, KNN, Naive Bayes, Random Forest**

I. Introduction

Real-world data will always have some degree of error. In images, these errors appear as noise, being visual distortions of the image's pixels. Popular datasets used in analyses of image classification model performance often do not reflect this reality [1] [2] [3]; they are cleaned, pre-processed, and free of noise. As such, these analyses make observations about the classifier's performance that are not necessarily consistent with how they would perform on real-world data. While the effect of noisy data has been studied extensively [4] [5], there is a major gap in knowledge pertaining to the effect of visual noise on image classifier performance; very few studies exist that analyze this subject. Various studies have analyzed the noise robustness of deep learning models [6] [7], but few studies evaluate the same topic for classic machine learning algorithms [8]. Further, this existing research for classic machine learning is limited to a handful of models, so there is insufficient information surrounding most popular image classifiers. Considering this, in this paper we dive into the unknown and evaluate the effect of visual noise on popular classic machine learning algorithms as image classifiers. The limited available facts make it difficult to form an explanation for any possible outcomes. As such, we ask the following question to guide this research: Of four common image classifiers [Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Random Forest], which ones produce the most desirable results when evaluating noisy images?

II. Motivation and objectives

The popularity of deep learning models in computer vision has skyrocketed in recent years, leading to a series of advancements in image classification that traditional machine learning models were not capable of [9]. Deep learning models have become the preferred method for image classification, as any simple google search will tell you. However, this does not negate the various use cases in which traditional machine learning models are preferred [10]. Despite this, further advancements in research and the performance of traditional machine learning models have been effectively pushed to the backburner. Very few studies exist that examine the effect of noise on machine learning models [8], much less any comparative studies for various models. This major gap in research has motivated us to conduct a comparative analysis on the effect of noise, specifically on common traditional machine learning algorithms in image processing.

The main objective, as previously stated, is to understand the effect of visual noise on various traditional machine learning models in image processing. To achieve this, we will apply different types of noise at varying levels on the images to obtain a full understanding of how each model reacts specifically to noise. Then, we will conduct a comparative analysis of each model to understand how they compare. The focus lies not on the effect of each type of noise specifically, but on how the model reacts to noise in general. Including various types of noise allows us to identify a model's behavioural trends easier.

III. Methodology

A. Python Tools and Libraries

Five main libraries were used to aid in the development of the Python code. Scikit-learn was used to create and tune the models, as well as split data and obtain metrics. Scikit-image was used to generate Impulse and Gaussian noise. Cv2 was used to generate image Blur. Matplotlib aided in visualising results. Then, Pandas and Numpy were utilized to convert images to flattened arrays.

B. Datasets

The Caltech-101 dataset consists of 9000 images across 101 classes [11], and has gained popularity in computer vision studies in recent years [12] [8]. Our study utilizes a subset of this dataset which contains 1200 images across 12 classes, which we have named “Caltech-101-1200”. Specifically, it consists of the first 100 images from all classes that have at least 100 images. The decision to use this subset stems from the issues presented by using the entirety of the Caltech-101 dataset. The foremost issue was the size of the dataset, as we did not possess the computing power to process and store all 9000 images in an acceptable amount of time. The second issue presented was the variance in class sizes, with some classes containing hundreds of images and others having less than 40 images. This variance would be appropriate were we to be analyzing these models' performance solely as image classifiers, but it does present issues when obtaining and understanding metrics related to the models' performance with noisy images.

30 noisy versions of Caltech-101-1200 were used; 10 versions for each type of noise generation, with each version representing a different severity level of noise.

C. Algorithms

1) SVM: classifies images by finding the largest hyperplane between the different classes. Hyperparameters of interest are a) regularization parameter (*C*) which describes the trade-off between maximizing margins and minimizing training errors, working to prevent overfitting, b) gamma (γ) defines the range of the points that are used to define the decision boundary, determining the extent of influence that each training example has on the model, and c) kernel (*K*) which determines the algorithm that is used to transform the input data for processing.

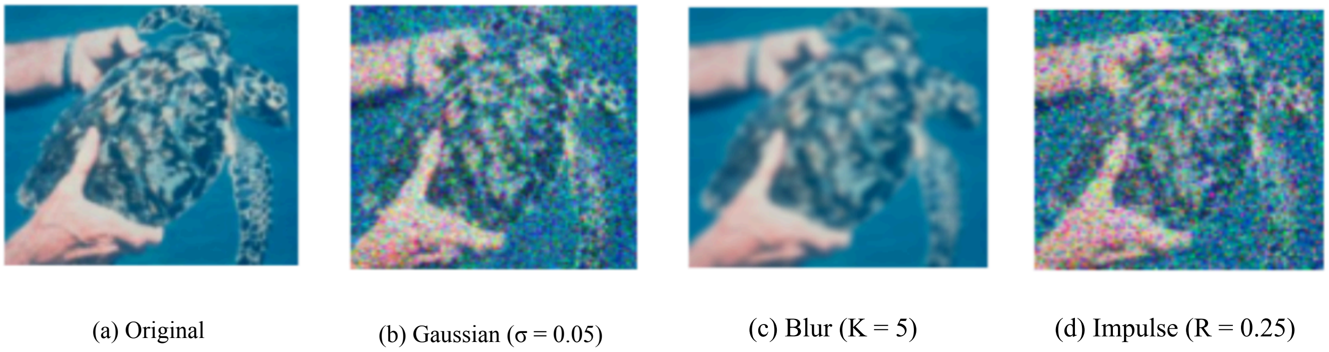
2) KNN: plots each image on a plane, then classifies the image based on the majority classification of its *k* closest neighbours; it has one main hyperparameter, the number of neighbours (*k*), which is the number of neighbours that the algorithm consults in order to make a classification decision.

3) Naive Bayes: applies Bayes theorem to classify images; the simplicity of the algorithm leads to a very minimal parameter set, meaning there are no hyperparameters of interest.

3) Random Forest: creates a number of decision trees to classify images based on the most common classification out of each decision tree; it is largely defined by its hyperparameters a) number of estimators (*n*) which describes the number of decision trees to be created, b) criterion (*c*) employs a criteria to determine how a decision tree split it made, c) maximum depth (*max*) defines the total number of splits each tree may make, and d) maximum features (*m*) which determines the total number of features that may be considered in each tree.

D. Noise

Three types of artificial noise were applied to the image set: Gaussian (Fig 17.b), Impulse (Fig 17.d), and Blur (Fig 17.c). Gaussian noise occurs in real-world situations from poor lighting conditions when an image is captured; it is generated based on Gaussian distribution, a probability density function. Its strength is measured by its standard deviation σ, and altering its strength will cause more or less variance in the change in pixels. Impulse noise occurs from bit errors typically caused by hardware malfunctions, and is generated by corrupting and replacing pixels in the original images. Its strength is measured by


(a) Original

(b) Gaussian (σ = 0.05)

(c) Blur (K = 5)

(d) Impulse (R = 0.25)

Fig. 17: Example image from Caltech-101-1200 with the various types of noise.

the frequency or rate of noise, *R*, which determines the probability of each pixel's corruption [13]. Finally, Blur noise occurs in real-world images for a variety of reasons, and causes objects in the image to appear fuzzy or distorted; it is generated through a kernel convolution, where each pixel's value is set to the mean of all pixels in the kernel. Its strength is determined by the kernel size, *K*, where Blur noise increases as *K* increases.

E. Metrics

To evaluate and understand the performance of each model, we decided on implementing three different yet similar metrics: accuracy, f-measure, and Equalized Loss of Accuracy. Accuracy is the classifier's correctness with prediction classifications, with a goal of approaching 100%. F-Measure depicts the classifier's ability to predict outliers, taking into account data distribution; the goal is to approach 1.0, indicating no false positives or false negatives. Equalized Loss of Accuracy (ELA) [14] measures robustness by determining how much accuracy is lost with the addition of noise; the closer to 1.0 ELA is, the greater the decrease in accuracy [15].

F. Parameter Initialization Methods

1) Noise Generation: The three types of artificial noises were applied to the image-set at various levels. For the Gaussian noise, we applied 10 values of standard deviation (σ = {0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1}). Similarly, for Impulse noise, 10 values of R were applied (R = {0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50}). Finally, for Blur noise, 9 values of K were applied (K = {2, 3, 4, 5, 6, 7, 8, 9, 10}).

2) Train/Test Split: To ensure reproducibility and equality throughout the experiment, we included a variety of parameter specifications to split the data into train and test sets. First, we applied an 80%/20% train/test split. Then, we used the random_state variable to ensure the split was the same for each dataset. Finally, we applied stratified sampling to ensure both the training and testing set had equal proportions of class labels.

3) Metrics: F-measure was calculated globally rather than through weighted or label-specific means, as we were working with a balanced dataset.

G. Experimental Process

1) Data Pre-Processing: There was minimal pre-processing of the image data needed. To pre-process the images, we (1) selected the subset of images used, such that the first 100 images from all classes with at least 100 images were included, (2) resized each image array to 150 x 150 x 3 to ensure consistency, (3) generated the 30 noisy versions of the dataset using the discussed methods and parameters, and (4) flattened all images for memory optimization.

2) Building Models: With regard to building the models, we followed a basic methodology for each algorithm. We (1) split the data into train/test sets using discussed parameters for images at 0% noise, then (2) trained the model on the 0% noise train image set with GridSearchCV (and appropriate hyperparameter grid, discussed further in Results) for hyperparameter tuning.

3) Obtaining Metrics: To obtain performance metrics for each model and the various levels of noise, we followed a basic methodology for each type and level of noise. We (1) split the data into train/test sets using discussed parameters for images of each type and level of noise (with each split only containing one level of noise), then (2) had each classify every version/level of the test set, and finally (3) obtained accuracy and f-measure for each classification task, and calculated ELA based on accuracy results.

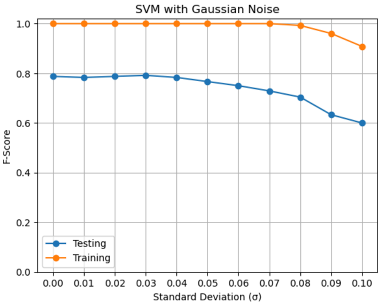


Fig. 1: SVM performance with Gaussian Noise.

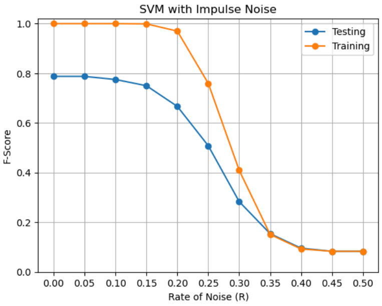


Fig. 2: SVM performance with Impulse Noise.

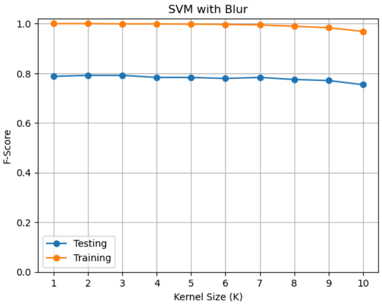


Fig. 3: SVM performance with Blur.

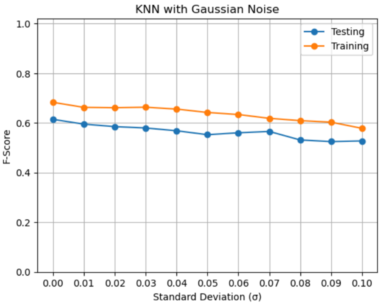


Fig. 4: KNN performance with Gaussian Noise.

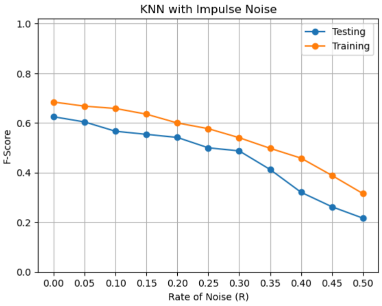


Fig. 5: KNN performance with Impulse Noise.

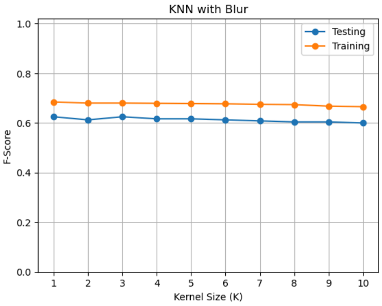


Fig. 6: KNN performance with Blur.

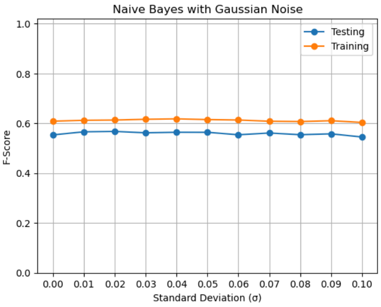


Fig. 7: Naive Bayes performance with Gaussian Noise.

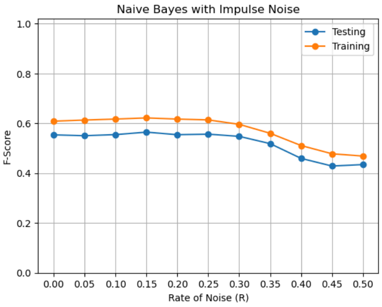


Fig. 8: Naive Bayes performance with Impulse Noise.

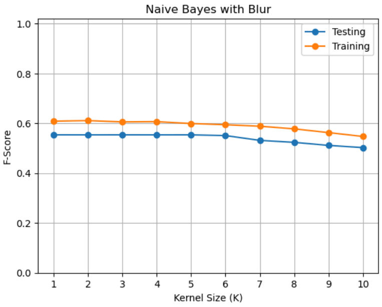


Fig. 9: Naive Bayes performance with Blur.

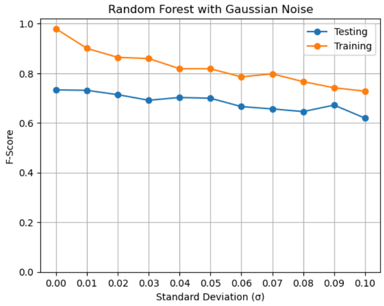


Fig. 10: Random Forest performance with Gaussian Noise.

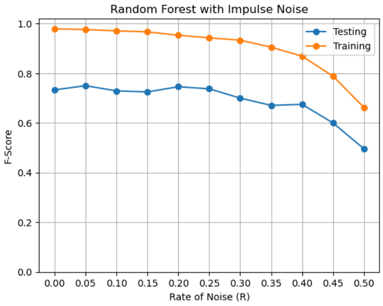


Fig. 11: Random Forest performance with Impulse Noise.

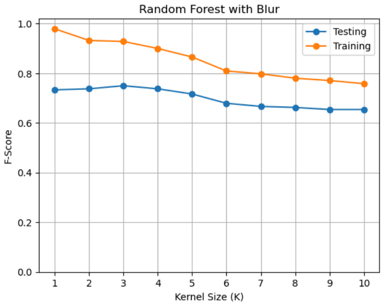


Fig. 12: Random forest performance with Blur.

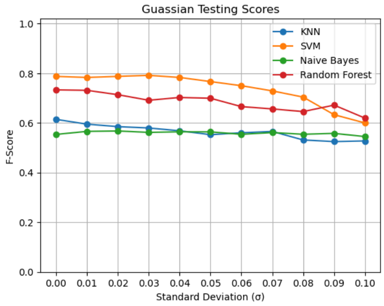


Fig. 13: Comparison of testing F-score results for images affected by Gaussian Noise.

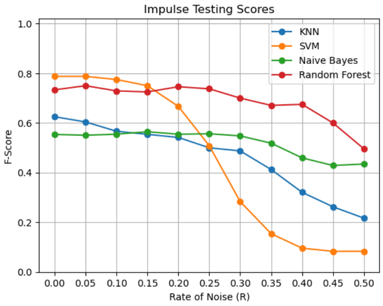


Fig. 14: Comparison of testing F-score results for images affected by Impulse Noise.

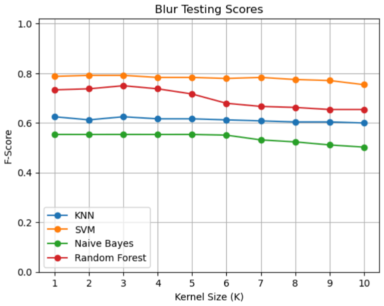


Fig. 15: Testing performance on Blur for all models.

Models	**Noise Types**		
	Gaussian Noise	**Impulse Noise**	**Blur**
SVM	0.5079	1.1640	0.3121
KNN	0.76	1.253	0.64
Naive Bayes	0.822	1.022	0.874
Random Forest	0.511	0.6875	0.4715

Fig. 16: ELA results for all models on testing data based on the highest level of noise.

IV. Results

Note that figures contain only F-Score as the values of accuracy and F-score were almost identical.

A. *Hyperparameters*

Hyperparameter tuning was conducted for each model through a grid search. For each model, the list of tried parameter values is provided, with the optimal tuned parameter bolded. In SVM, 5 values of regularization (C = {0.1, **1**, 10, 50, 100}), 4 values of gamma (γ = {**0.0001**, 0.001, 0.01, 1}), and 3 different kernels (K = {**rbf**, poly, linear}) were applied. In KNN, 15 different numbers of neighbours were applied (k = {1, 2, 3, 4, 5, 6, 7, **8**, 9, 10, 11, 12, 13, 14, 15}). In Random Forest, 11 different numbers of estimators (n = {100, 105, 110, 115, 120, 125, **130**, 135, 140, 145, 150}), 2 types of criterions (c = {**entropy**, gini}), 4 values of maximum depth (max = {5, 6, **7**, 8}), and two types of considerations for maximum features (m = {**sqrt**, log2}) were applied.

B. Algorithm Performance

1) SVM: With all types of noise, the overall trend was that as the level of noise increased the F-score decreased (Fig. 1,2,3). Impulse noise had the largest decrease in both training and testing F-score, dropping . Blur and Gaussian noise did not cause the F-score to strictly decrease, with some levels increasing the accuracy slightly (Fig. 1,2).

2) KNN: KNN generally decreased in testing and training F-score as the level of noise increased (Fig. 4,5), except for Blur noise, which remained stable (Fig. 6). Impulse noise decreased the performance the most, followed by Gaussian noise. (Fig. 4,5).

3) Naive Bayes: With all types of noise, Naive Bayes' F-score fluctuated up and down as noise increased (Fig. 7,8,9). Impulse noise and Blur had an overall downward trend, with Impulse causing the scores to decrease more (Fig. 8,9). Gaussian noise had no clear trend, with the original and noisiest scores being within 0.01 of each other (Fig. 7)..

4) Random Forest: With all types of noise, Random Forest had a downward trend as the noise intensity increased (Fig. 10, 11, 12). As well, all types of noise caused the testing F-score to increase at certain levels (Fig. 10, 11, 12). Impulse noise caused the largest decrease in F-score for both training and testing (Fig. 11).

C. Comparative Analysis Results

For Gaussian noise, all models remain somewhat stable. At the highest noise levels, KNN had the lowest F-score, while SVM had the largest decrease, and Random Forest had the second largest decrease. Naive Bayes had the smallest change in F-score, with the highest noise F-score being within 0.01 of the original data F-score.(Fig. 13).

With Impulse noise, all models had an overall downward trend, with SVM being the only one that is strictly downwards. SVM had the largest drop in F-score when comparing the noisiest data to the original data. KNN had the second biggest drop. Naive Bayes had the smallest change in its score overall, and remained the most stable (Fig. 14).

Blur caused similar results across the model, except for Random forest, which had the largest increase and decrease in F-score. KNN and SVM both were somewhat stable and had a relatively small difference between their F-scores. Naive Bayes remained stable for the first few levels of noise, before lowering at a rate smaller than Random forest. (Fig. 15).

V. Discussion

A. Findings

With regards to the different types of noise, it is clear that Gaussian and Blur have minimal impact on performance, while Impulse noise results in a major loss of accuracy for most of the models. As such, a model's behaviour when evaluating Impulse noise provides an exaggerated understanding of the algorithm's robustness.

SVM has high accuracy under the right conditions, but reacts very poorly to Impulse noise, having a more than 80% loss in accuracy. Additionally, the consistent 20% disparity in testing and training values signifies intense overfitting. This disparity resolves itself in the later half of Impulse noise levels, though the poor performance that occurs alongside it indicates major underfitting.

KNN performs generally around or slightly above average and is generally stable, but experiences a ~40% drop in accuracy when evaluating Impulse noise.

Naive Bayes has an accuracy and F-Score that is constantly around or slightly above average, and is consistently stable throughout all types and levels of noise.

Random Forest consistently performs far above average, though responds similarly to all types of noise with at most a 20% drop in accuracy. The >20% disparity in training and testing values indicates an issue of overfitting.

B. Interpretation of Results

Here we aim to use an analysis of the experiment's results, in tandem with existing research, to answer the proposed question: Of four common image classifiers [Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Random Forest], which ones produce the most desirable results when evaluating noisy images? Note that existing research referenced is generally focused on each algorithm's performance with noisy data rather than noisy images.

Results related to SVM indicate that the classifier is not sufficiently robust to noise. While it performs relatively well with Gaussian and Blur noise (having a testing accuracy within the range of 60-80%), its incredibly poor performance with Impulse noise shows that its performance is too inconsistent to be a reliable classifier. This analysis is corroborated by Paranhos da Costa et al. [8], which identified a similar drastic loss in accuracy for SVM when classifying images with Poisson noise. Further, the signs of having simultaneous issues of overfitting and underfitting make it an inappropriate classifier to use for noisy images.

KNN is insufficiently robust to noise as well. While it performs above average as a classifier generally, its drastic loss in accuracy with Impulse noise specifically is a clear indicator that it is not fit to perform classification on noisy data. This is also indicated simply by the nature of the algorithm, as its process makes itself more susceptible to noise [5].

Naive Bayes is consistent among all types of noise, remaining robust even against Impulse noise. As well, it shows no signs of over or underfitting. These results are consistent with general understanding of Naive Bayes' performance as a classifier [4]. The model's robustness to noise is substantial, so much so that its performance at higher levels of noise is sometimes greater than that at the lowest level of noise. Such a phenomenon has produced incoherent ELA results, as it is consistently close to 1.0 (which normally indicates a greater loss in accuracy). Despite the model's desirability, its performance (that being, its accuracy) is not notable, sitting in the range of ~40-60%. Since it has no hyperparameters to tune, it would be difficult to increase this accuracy without altering the dataset itself.

Random Forest is also very robust to noise, as shown by its relatively lower ELA scores. Its training accuracy is consistently the highest of all the models for all types of noise, and its testing accuracy is consistently the second highest. However, despite its accuracy consistently outperforming the other models, Random Forest still suffers from a great loss of accuracy at higher levels of noise. This renders its ELA values incomprehensible, as they are the lowest of any algorithm, but the model itself experiences such a great loss. Though, regardless, it maintains a very high accuracy even at the highest levels of noise. Another issue of concern is its signs of overfitting (though typical for Random Forest [16]), which makes it more undesirable as a classifier.

Overall, Random Forest and Naive Bayes are the only models in this study that are robust to noise. Each of them have their own issues and merits that must be weighed when considering which to use. Naive Bayes offers greater consistency, while Random Forest presents a higher accuracy.

C. Pros and Cons of Implemented Solution

Many aspects of our experiment have provided us with a sufficient understanding of the results. By implementing multiple types of noise, we were able to fully grasp what behaviour should be considered “normal” for a certain type, and thus identify any models that presented abnormalities (e.g. SVM behaviour with Impulse noise). Additionally, the use of a smaller dataset allowed us to obtain a better understanding of each model's accuracy, having its starting accuracy at 0% be high or low enough to fully understand how the model behaviour reacts and changes with the addition of noise. However, the small dataset may lead to the results being not wholly representative of the classifiers’ performance on a real-world dataset. Further work can include experimenting on a larger, non-linear dataset.

VI. Conclusion

The prominence of visual noise in real-world data is of increasing interest in image classification problems. However, there is a major gap in knowledge in this area, due to the lack of research regarding the effect of visual noise on classic machine learning algorithms. The results and analysis provided in previous sections have helped to bridge this gap by identifying noise-related behaviour for popular machine learning image classifiers.

Results from this study determined the robustness of SVM, KNN, Naive Bayes, and Random Forest by applying varying levels of Impulse noise, Gaussian noise, and Blur noise. Results made clear that SVM and KNN are simply not robust against noise, while Naive Bayes and Random Forest are. Naive Bayes and Random Forest have their own issues and merits to be considered, and future works could further analyze these classifiers differences to determine which is most desirable.

Regardless of robustness, each model is affected by noise in general, though different types of noise have varying effects on each model. It’s important to explore and understand these effects in order to apply these models more effectively

References

[1] P. Wang, E. Fan, and P. Wang, “Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning,” Pattern Recognition Letters, vol. 141, Aug. 2020, doi:10.1016/j.patrec.2020.07.042

[2] M. A. Abu, N. H. Indra, A. H. A. Rahman, N. A. Sapiee, and I. Ahmad, “A study on Image Classification based on Deep Learning and Tensorflow,” International Journal of Engineering Research and Technology, vol. 12, no. 4, pp. 563–569, 2019.

[3] M. Hussain, J. J. Bird, and D. R. Faria, “A Study on CNN Transfer Learning for Image Classification,” research.aston.ac.uk, Aug. 11, 2018. doi:10.1007/978-3-319-97982-3_16

[4] R. H. Stribos, "The Impact of Data Noise on a Naive Bayes Classifier." University of Twente, 2021. [Online]. Available: https://purl.utwente.nl/essays/85678

[5] S. Ougiaroglou and G. Evangelidis, “Dealing with noisy data in the context of K-NN classification,” Proceedings of the 7th Balkan Conference on Informatics Conference, 2015. doi:10.1145/2801081.2801116

[6] H. Jang, D. McCormack, and F. Tong, “Noise-trained deep neural networks effectively predict human vision and its neural responses to challenging images,” PLOS Biology, vol. 19, no. 12, 2021. doi:10.1371/journal.pbio.3001418

[7] M. Momeny, A. M. Latif, M. Agha Sarram, R. Sheikhpour, and Y. D. Zhang, “A noise robust convolutional neural network for Image Classification,” Results in Engineering, vol. 10, p. 100225, 2021. doi:10.1016/j.rineng.2021.100225

[8] G. B. Paranhos da Costa, et al., "An empirical study on the effects of different types of noise in image classification tasks," 2016. [Online]. Available: https://arxiv.org/abs/1609.02781

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.

[10] “Decoding the dichotomy: Traditional Image Processing vs. Deep Learning Whitepaper.” [Online]. Available: https://www.imveurope.com/sites/default/files/content/white-paper/pdfs/HCL_IMVE_WP-ImageProcessing_vs_DL.pdf

[11] Li Fei-Fei, R. Fergus and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," 2004 Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, 2004, pp. 178-178, doi: 10.1109/CVPR.2004.383.

[12] Dang, Y., Jiang, N., Hu, H. et al. Image classification based on quantum K-Nearest-Neighbor algorithm. Quantum Inf Process 17, 239 (2018). doi:10.1007/s11128-018-2004-9

[13] A. Awad, “Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise,” Engineering Science and Technology, an International Journal, vol. 22, no. 3, pp. 746–753, Jun. 2019, doi: https://doi.org/10.1016/j.jestch.2019.01.012.

[14] J. A. Sáez, J. Luengo, and F. Herrera, “Evaluating the classifier behavior with noisy data considering performance and robustness: The equalized loss of accuracy measure,” Neurocomputing, vol. 176, pp. 26–35, 2015. doi:10.1016/j.neucom.2014.11.086

[15] D. Ljunggren and S. Ishii, "A Comparative Analysis of Robustness to Noise in Machine Learning Classifiers," KTH Royal Institute of Technology, 2021 [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1597519/FULLTEXT01.pdf

[16] M. Sheykhmousa, et al., “Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 6308–6325, 2020, Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9206124&tag=1