RE: Active Negative Loss Functions for Learning with Noisy Labels

Briksam Kasimoglu *briksam.kasimoglu@bahcesehir.edu.tr*
Student Number: 2102969

Sara Amhan *sara.amhan@bahcesehir.edu.tr*
Student Number: 2105687

Abstract

This reproducibility report focuses on verifying claims stated in the original paper [11] by re-conducting their experiments using the provided codebase and comparing results. The original paper focused on satisfying the need to find an alternative passive loss function that is robust to noisy label learning to replace MAE in a previous state-of-the-art framework Active Passive Loss (APL) since MAE has limited fitting ability when it comes to complex datasets and takes a long time to converge during training. The authors proposed a new passive noise-robust loss function class named Negative Normalized Loss functions (NNLFs) as a replacement and by doing so they introduce a novel framework called Active Negative Loss (ANL), the framework improves convergence speed, accuracy, and overall performance when compared to other loss functions and state of the art methods. We have reproduced most of the original paper's results, our work can be found *here*.

1 Reproducibility Summary

Scope of Reproducibility — In this work We examine the reproducibility of the paper "Active Negative Loss Functions for Learning with Noisy Labels" to validate the main claims stated by the authors as the following: **1)** The passive loss NNLF and the framework ANL are robust to noisy labels. **2)** NNLFs show better fitting ability than MAE on complex datasets. **3)** L1 regularization helps with overcoming the ANL framework's overfitting problem. **4)** While NNLFs are robust to noise and show good performance, at very high noise rates a combination of active and passive losses following the ANL framework is needed to get better results. **5)** The ANL framework surpasses the state-of-the-art methods with its performance on benchmark datasets.

Methodology — To conduct our experiments, we reused the author's code with some adjustments. The modifications were applied to fix some errors and run experiments that were mentioned in the original paper but had no related code in the official repository. We ran the experiments on various platforms, including Google Colab, Kaggle, and CPUs, which took 207 hours of training. For a more detailed explanation, please refer to ***Computational Requirements*** in Section 4.3.

Results — Claim 1, 2, and 3 are verified in section 5.1 by following the clean and noisy training set accuracies in the graph we can verify the noisy label robustness on two datasets and compare them to the other proven non-robust method (Cross-Entropy), ANL surpassing the fitting ability of MAE claim was proved by the inability of MAE to fit the CIFAR-100 [5] dataset, and the L1 regularization claim was proved by the plot generated on the two datasets. Claim 4 is verified in Section 5.2, by comparing the performance of NNCE alone versus when it's implemented in the ANL framework makes it clear that following the ANL framework is needed to achieve a better result. Lastly, claim 5 is verified in section 5.3 as the proposed method's test accuracy surpasses all other methods on different datasets, noise types and noise rates.

What was easy — The paper was very detailed, simple to follow and understand. Code was available and had TensorBoard integrated, although it needed some modifications. For more details, please check out Section 6 ***What was easy***.

What was hard — We were mainly having a hard time dealing with the limited computational resources as the available online platforms have limited quotas which makes it hard to sustain a very long training session and local resources were limited to CPUs so running a large dataset would result in days of waiting. For more details, please check out Section 6 ***What was hard***.

Communication with original authors — We were able to contact one of the authors, however, it was unfortunately a late reply. For more details, please check out Section 6 ***Communication with original authors***.

2 Introduction

One of the many problems faced when deep neural networks (DNNs) are used for supervised classification tasks is noisy label learning. DNNs achieve good results when provided with meticulously annotated large-sized data, but when it comes to real-world applications, the process of neatly and correctly labeling that amount of data is an expensive and usually erroneous process which brings mislabeled datasets to the equation. Studies show how easily DNNs can fit a dataset with random labels [13] and that can lead to a low evaluation performance when the training set is noisy. Consequently, the interest in finding a solution to the noisy label learning problem led to the search for loss functions that are robust to noise, Ghosh et al. [2] have shown that symmetric loss functions like Mean Absolute error (MAE) are noise-robust unlike Cross Entropy (CE) and others. Even though it's robust MAE takes a longer time to train before it converges because it treats all the samples equally which makes the learning process harder suggesting that MAE is not the best pick for complex datasets [14]. To find a better pick, many partially robust loss functions were proposed such as Generalized Cross Entropy (GCE) [14], Symmetric Cross Entropy (SCE) [9] , and Reverse Entropy (RCE) which are all variants of CE and MAE. Finally, the Active Passive Loss (APL) [7] framework was proposed to make any loss function fully robust simply by applying a normalization operation that makes them symmetric. This framework classifies the normalized loss functions into two types: "Active" and "Passive" loss, which mutually boost one another to mitigate the underfitting problem of normalized loss functions, however, the passive losses used by APL are all scaled MAEs and as mentioned previously MAEs are not good for convergence. The authors' goal is to create a loss function to replace the MAE in APL to optimize prediction results on noisy datasets.
In this report, we take the authors' proposed new class of robust passive loss functions Normalized Negative Loss Functions (NNLFs), and the improved APL framework named Active Negative Loss (ANL) and check their effectiveness by doing the experiments done in the original paper which include the comparison of loss functions on datasets with various noise rates and we compare our results and report on the details and obstacles faced during this process.

3 Scope Of Reproducibility

The main objective of the original paper is to propose a passive loss function that is robust to noisy labels to replace the MAE loss in the passive component of the APL framework. The authors propose a new class of noise-robust passive loss functions called Normalized Negative Loss functions (NNLFs) and use them to define a modified framework named Active Negative Loss (ANL).
In this study we aim to verify the following claims of the original paper:

1. The passive loss NNLF and the framework ANL are robust to noisy labels tested on different datasets with two types of noise to the labels symmetric and asymmetric, and different noise rates.

2. NNLFs show better fitting ability than MAE on complex datasets.

3. L1 regularization helps with overcoming the ANL framework's overfitting problem.

4. While NNLFs are robust to noise and show good performance, at very high noise rates a combination of active and passive losses following the ANL framework is needed to get better results.

5. The ANL framework surpasses the state-of-the-art methods with its performance on benchmark datasets such as MNIST, CIFAR-10, and CIFAR-100. Tested by applying the framework on Cross-Entropy and Focal loss, the loss functions ANL-CE and ANL-FL are compared to other loss functions.

Claims one through four are verified by re-running the experiments done in the original paper through training networks that use each one of the 9 loss popular functions, two of which are the proposed method (ANL-CE and ANL-FL). The process is done on 3 different datasets. Each network is trained on clean data, data with symmetric noise rates η (.4, .6, .8), and data with asymmetric noise rates η (.2, .3, .4). This experiment will produce 189 corresponding accuracies and those will be used to compare the behavior of the proposed method on the benchmark datasets.

4 Methodology

The paper proposes a new loss function framework to improve the Active Passive Loss (APL) [7] (Framework from a previous study), The new framework uses different forms of the same active loss function to implement the active and passive parts, for the active part they use a normalized active loss function (symmetric and robust to noise) and for the passive part they implement their new class of robust passive loss functions namely Normalized Negative Loss functions (NNLFs) which can be created by taking any active loss function and combining 1) complementary label learning [4, 12] and 2) a simple vertical flipping operation, this way all active loss functions can be made into passive ones. For instance, when it comes to Cross Entropy (CE) when implemented in the ANL framework the new loss function is as such:

$$\mathcal{L}_{ANL-CE} = \alpha \cdot \mathcal{L}_{NCE} + \beta \cdot \mathcal{L}_{NNCE}$$

For reproducing the results, we used the author's original code, modifying some content because of some errors we ran into and adding code files to get the graphs and some results whose configuration files were not provided in the original repository [11]. We found the paper to be fairly easy to follow, and the code, despite the lack of some necessary files, was easy and simple to understand. In the coming subsections, explain the datasets, models, and experiments we have done in our study.

4.1 Datasets

The datasets we used are the ones used in the original paper: MNIST [6](12), CIFAR-10 [5](13), and CIFAR-100 [5](13). In the experiments, some noise is applied to the dataset in question to investigate the performance of the respective method. There are two types of noise applied in the experiments: symmetric noise and asymmetric noise. The original paper follows standard approaches in previous works [8, 7, 15] for generating the noisy labels. For symmetric noise, the labels are flipped in each class randomly to match the incorrect labels of other classes. For asymmetric noise, the labels are flipped within a specific set of classes. For symmetric noise, the noise rate is $\eta \in \{0.2, 0.4, 0.6, 0.8\}$ (for some experiments, we only use a subset because of a lack of computing power), and for asymmetric noise, $\eta \in \{0.2, 0.3, 0.4\}$.

MNIST [6] — (Modified National Institute of Standards and Technology) is a set of handwritten digits that includes 60,000 training samples and 10,000 testing samples, all in a fixed size of 28×28 pixels and in grey scale. For noise application, the labels are flipped as the following: $7 \rightarrow 1$, $2 \rightarrow 7$, $5 \leftrightarrow 6$, and $3 \rightarrow 8$.

CIFAR-10 [5] — (Canadian Institute For Advanced Research) is a subset of the 80 million tiny images dataset with independently generated labels. The dataset contains 50000 32x32 colored training images in 10 classes and 10000 test images. For CIFAR-10, the flipping is: $TRUCK \rightarrow AUTOMOBILE$, $BIRD \rightarrow AIRPLANE$, and $DEER \rightarrow HORSE$, $CAT \leftrightarrow DOG$.

CIFAR-100 [5] — is the same as CIFAR-10 only with the classes being 100 instead of 10, with the 100 classes being grouped into 20 super-classes with each having 5 sub-classes [3]. For CIFAR-100, to apply noise to classes, each class is flipped within the same super-class into the next class in a circular fashion.

4.2 Models

Followed from a previous study [7, 15]. ResNet34 [3] was trained for 200 epochs, ToyModel4L for 120 and ToyModel8L for 200. All the training were ran with an SGD optimizer, 0.9 momentum and cosine learning rate annealing.

ResNet34 [3] — is a residual network that has 34 layers of convolutions with shortcut connections added to each pair of 3×3 filters. After the convolutions layers, a global average pooling layer reduces the spatial dimensions to 1x1 and then fed into a fully connected layer that outputs the class scores . The implementation of this network in the original paper was referenced to another paper [3]. We have used the same implementation and experimental setting as the original paper to train the model on CIFAR-100.

ToyModel4L — is a basic 4-layer convolutional neural network. The implementation in the original code consists of four building blocks, two of which are an application of a convolution operation followed by batch normalization and ReLU activation as a layer. Each of those two blocks is followed by a max pooling operation, and the last two layers are fully connected layers for the final classification. In our experiments, we used the same implementation on the MNIST dataset.

ToyModel8L — is also a basic 8-layer convolutional neural network. It has the same implementation as the ToyModel4L, but instead of the building blocks having one convolution layer followed by batch normalization and ReLU activation, it has two followed by max pooling. The implementation has three of the previous blocks and two fully connected layers. In total, there are six convolutional layers and two fully connected layers.

4.3 Experimental Setup

For the experiments in our reproduction study, we can divide the experiments into 3. In the first part of the experiment, we, as in the original paper, try to investigate the robustness to noisy samples and the fitting ability of the proposed method and compare it to other state-of-the-art methods using different noise rates and different-sized datasets (CIFAR-10 & CIFAR-100). In the second experiment, we try reproducing the results of the original paper to support their claims, which signify that at a very high noise rate, a combination of active losses is needed to achieve better performance. To do that, we run the normalized loss function and investigate its behaviour, then we run the normalized negative same loss function to make it a robust passive loss function and again investigate its behaviour, and then we run the same loss function in the ANL framework and again investigate its behaviour. This experiment is done on CIFAR-10. In the third and last experiment, we run different state-of-the-art loss functions and the proposed novel framework on different datasets and noise rates and examine the performance of each method.

Hyperparameters — In our reproduction study, we used the same hyperparameters in all experiments except for the first experiment. Unfortunately, the first experiments' hyper-parameters were not provided by the original authors, so we had to navigate through the code and make our own configuration file. In the configuration file, we tried using the most commonly used values in other similar methods that use the same hyperparameters. As a result, the hyper-parameters for the first experiment are: a) For Normalized Cross Entropy (NCE): SGD optimizer, 0.01 learning rate, 0.9 momentum, 1×10^{-4} weight decay, and 5.0 gradient bound, which are the same in almost all the experiments/ trainings; b) Normalized Negative Cross Entropy (NNCE): all previously mentioned in NCE with β (Normalized Negative Loss function weight) being 5.0, δ (L1 regularization term/weight) being $5e-5$, and minimum probability $= 1 \times 10^{-7}$ c) For Active Negative Cross Entropy, the hyper-parameters are all the same throughout the experiment. For the rest of the experiments, we used the same hyper-parameters as the original authors, which are shown in Table [1].

Computational Requirements — We ran the experiments on the following libraries: Python 3.10, Pytorch 2.3.0, Torchvision 0.18.0, and Numpy 1.26.2. As for the computational resources, we used Kaggle, Google Colab, two local CPUs (AMD RYZEN 7 and Intel i5 11th Gen).It took us 207 hours of training in total (both locally and online), the previously mentioned hours only encompass the running times not the full hours spent working on the report.

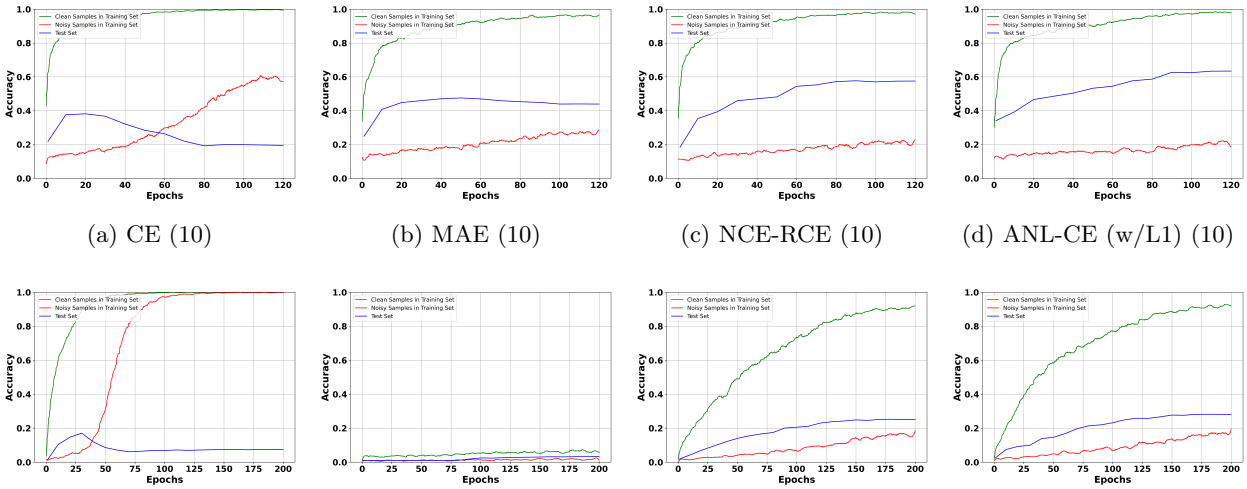
Table 1: Hyper-parameters settings for different methods in (most) the experiments. ((-) means no hyper-parameters, - means experiment not down)

Method	MNIST	CIFAR-10	CIFAR-100
CE (-)	-	(-)	(-)
MAE (-)	(-)	(-)	(-)
SCE [9] (α, β)	-	(0.1, 1.0)	-
NCE+RCE [7] (α, β, γ)	(1.0, 10.0)	(1.0, 1.0)	(10.0, 0.1)
NCE+AGCE [15] (α, β, a, q)	(0.0, 1.0, 4.0, 0.2)	(1.0, 4.0, 6.0, 1.5)	(10.0, 0.1, 1.8, 3.0)
ANL+CE (α, β, δ)	(1.0, 1.0, 1×10^{-6})	(5.0, 5.0, 5×10^{-5})	(10.0, 1.0, 5×10^{-7})
ANL+FL (α, β, δ, γ)	(1.0, 1.0, 1×10^{-6}, 0.5)	(5.0, 5.0, 5×10^{-5}, 0.5)	(10.0, 1.0, 5×10^{-7}, 0.5)

5 Results

5.1 Result 1

To generate the training and test accuracy plots in Figure [1] we train with the following loss functions 1) CE, 2) MAE, 3) NCE+RCE [7], and 4) ANL-CE (w/ L1) on two datasets CIFAR-10/-100 [5] with the symmetric noise rate 0.8 for both datasets and compared our results: In general, all the plots have the same shape as the plots in the original paper which supports the reproducibility of the paper. Many observations can be made from the generated plots that support the claims we are trying to verify as the following: 1) Replacing MAE in the APL framework gave better performance on the test set as can be seen in subgraph (1d) and (1h) (ANL) versus the plots (1c) and (1g)(APL), and plots (1b) and (1f) representing MAE itself. 2) In subgraph (1f) CIFAR-100 we can see how MAE is not able to fit the complex data even though its performance on CIFAR-10 shows robustness to noise, ANL shows better fitting ability on CIFAR-100 (1h). 3) In ANL the difference between the clean sample accuracies and the noisy sample accuracies of the training set is maintained which indicates robustness to noise unlike CE. 4) Using L1 as a regularization method helps with overcoming the overfitting problem of ANLs.



(a) CE (10) (b) MAE (10) (c) NCE-RCE (10) (d) ANL-CE (w/L1) (10)

(e) CE (100) (f) MAE (100) (g) NCE-RCE (100) (h) ANL-CE (w/L1) (100)

Figure 1: Training and test accuracies of different loss functions. (a) - (d) are training accuracies on CIFAR-10 with 0.8 noise rate. (e) - (h) are training accuracies on CIFAR-100 with 0.8 noise rate. Low noisy sample training accuracy shows how much robustness the method has.

5.2 Result 2

As seen in Table [2], the two components of ANL-CE: Normalized Cross Entropy (NCE) (Symmetric-Active) and Normalized Negative Cross entropy (NNCE) (Passive) along with ANL-CE itself were trained and tested on CIFAR-10 for different noise rates ($\eta \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$) to show the effectiveness of each component alone. Some of the results (NCE and NNCE) showed differences with the original paper due to the fact that we set the hyperparameters of them ourselves and created the config file because it wasn't provided. We observe that while NNCE the NNLF part of the ANL framework is robust to noise and performs well enough in low noise rates, for higher noise rates ANL-CE is needed for achieving higher test accuracies which confirms the claim of the paper when it comes to the ANL framework.

Table 2: Different methods test accuracies (%) on CIFAR-10 for different noise rates ($\eta \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$). The best accuracy, for each noise rate η, is in **bold**.

Method	Clean ($\eta = 0.0$)	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$
NCE	75.09	72.41	69.25	63.44	40.79
NNCE	87.70	86.33	83.10	75.55	40.64
ANL-CE	**91.67**	**90.03**	**87.14**	**81.09**	**62.30**

5.3 Result 3

In this experiment, we considered different loss functions for three benchmark datasets which are: MNIST, CIFAR-10, and CIFAR-100. For each dataset, we evaluated the performance of different loss functions with different noise types and rates and organized them in two tables for each dataset one that has the proposed ANL framework and another that has the other loss functions including the latest state-of-the-art methods. Not all 189 accuracies were reproduced due to time and computational constraints, however, all results pertaining to the ANL framework were reproduced fully for the hyperparameters used please refer to Table [1]. For MNIST refer to Table [3], for CIFAR-10 refer to Table [4], and for CIFAR-100 refer to Table [5]. The results we get are very similar to the original Table, which again supports the reproducibility of this paper. From Tables [3, 4, 5] it can be observed that the ANL framework (ANL-CE and ANL-FL) performs better in almost all noise rates, however, there is a significant difference in performance when the noise rate is very high and/or the dataset is very complex in comparison with the previously studied APL and other loss functions.

Table 3: Different methods accuracies (%) on the MNIST dataset. The best accuracy, for each noise rate η, is in **bold**.

Datasets	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.8	Asymmetric Noise Rate (η) 0.2	0.4
MNIST	MAE	**99.24**	70.01	99.03	92.51
	NCE+RCE [7]	**99.46**	74.53	99.01	91.37
	NCE+AGCE [15]	99.00	**96.84**	**99.13**	88.68

Dataset	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.4	0.6	0.8	Asymmetric Noise Rate (η) 0.2	0.3	0.4
MNIST	ANL-CE	99.17	98.80	98.52	**96.49**	**99.17**	98.96	**98.09**
	ANL-FL	99.17	98.83	98.40	93.36	99.12	99.00	**98.13**

Table 4: Different methods accuracies (%) on the CIFAR-10 dataset. The best accuracy, for each noise rate η, is in **bold**.

Datasets	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.8	Asymmetric Noise Rate (η) 0.3	0.4
CIFAR-10	CE	91.07	19.49	78.40	73.84
	MAE	89.08	43.91	56.08	56.07
	SCE [9]	**91.68**	27.95	80.29	73.72
	NCE+RCE [7]	91.40	57.53	84.99	77.39
	NCE+AGCE [15]	90.77	47.67	85.52	**77.91**

Dataset	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.4	0.6	0.8	Asymmetric Noise Rate (η) 0.2	0.3	0.4
CIFAR-10	ANL-CE	**91.67**	87.14	81.09	**62.30**	89.40	**85.58**	77.39
	ANL-FL	91.55	87.10	82.18	**62.35**	89.07	**86.15**	**77.73**

Table 5: Different methods accuracies (%) on the CIFAR-100 dataset. The best accuracy, for each noise rate η, is in **bold**.

Datasets	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.8	Asymmetric Noise Rate (η) 0.4
CIFAR-100	CE	**71.19**	7.45	41.24
	MAE	5.72	3.31	5.65
	NCE+RCE [7]	68.47	25.17	43.21
	NCE+AGCE [15]	69.12	25.75	44.65

Dataset	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η) 0.4	0.6	0.8	Asymmetric Noise Rate (η) 0.2	0.3	0.4
CIFAR-100	ANL-CE	**70.30**	61.86	52.37	**28.10**	66.11	59.59	**45.24**
	ANL-FL	69.80	62.46	51.00	**28.34**	66.99	59.90	**46.09**

6 Discussion

In conclusion, we can confidently say that the original paper was well written and easy to follow to a certain extent there might be a need to check out the previous studies referenced in the original paper to get a better grasp on the theory. The official code repository provided was straightforward and easy to use with the exception of some missing configuration files and two to three errors that needed fixing, we were able to run almost all of the official experiments in the paper required to reproduce this report except for the experiment conducted on the real-world datasets Web-Vision [10] and ILSVRC-2012 [1] due to missing code and limited computational power. Overall, the reproduced results were very similar to the original paper with minor differences and mainly in support of the previously stated 5 claims. We would like to point out that when running the experiment in Table [3] on the MNIST dataset, the APL framework outperformed the newly proposed ANL framework in all noise rates except for the high asymmetric noise this was observed in both the original paper and our results, we assume that since MAE is a robust function and MNIST is a small enough dataset the APL framework (NCE+AGCE [15]) held its title as a state of the art method, and as stated in the original paper if the regularization method is improved ANL has the potential to achieve even higher results.

What was easy — The original paper had most of the theory well explained and provided mathematical breakdown and proofs in the appendix along with values for most hyperparameters. The official code repository provided was straightforward to understand and run as long as you know Python/ Pytorch basics, the three datasets used in our reproduced experiments were publicly available and automatically downloaded by the code. The code also utilized the Tensorboard library which made it easy for us to generate plots.

What was hard — The computational resources hardly sustained what we needed to be able to run the experiments continuously since we were limited to approximately 4 hours of GPU usage per day on Google Collab and 30 hours of weekly quota on Kaggle only, and the local CPUs were very slow to work on. Google Collab had constant runtime issues where it would crash halfway through the training, and we would lose all the output files generated whenever this happened. We had to write our own config files for some experiments and modify the code to run said files.
Communication with original authors — We tried contacting one of the authors for the Web-Vision dataset as we had to manually download it and it was large sized around 16 gigabytes, even after downloading it we ran into errors and we wouldn't have been able to run experiments except locally since the data is too big for online platforms, when we didn't get a reply we tried to contact the remaining authors and got a late reply to help fix the errors after downloading.

References

[1] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[2] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. "Robust Loss Functions under Label Noise for Deep Neural Networks". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 2017, pp. 1919–1925.

[3] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[4] Takashi Ishida et al. "Learning from Complementary Labels". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 5639–5649.

[5] Alex Krizhevsky and Geoffrey Hinton. "Learning multiple layers of features from tiny images". In: *2009*. 2009.

[6] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[7] Xingjun Ma et al. "Normalized Loss Functions for Deep Learning with Noisy Labels". In: *Proceedings of the 37th International Conference on Machine Learning*. 2020, pp. 6543–6553.

[8] Giorgio Patrini et al. "Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach". In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2233–2241.

[9] Yisen Wang et al. "Symmetric Cross Entropy for Robust Learning with Noisy Labels". In: *Proceedings of the 2019 IEEE International Conference on Computer Vision*. 2019, pp. 322–330.

[10] WenLi et al. "Webvision Database: Visual Learning and Understanding from Web Data". In: *CoRR* abs/1708.02862 (2017). URL: http://arxiv.org/abs/1708.02862.

[11] Xichen Ye et al. "Active Negative Loss Functions for Learning with Noisy Labels". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 6917–6940. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/15f4cefb0e143c7ad9d40e879b0a9d0c-Paper-Conference.pdf.

[12] Xiyu Yu et al. "Learning with Biased Complementary Labels". In: *Proceedings of the 15th European Conference on Computer Vision*. 2018, pp. 69–85.

[13] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *Proceedings of the 5th International Conference on Learning Representations*. 2017.

[14] Zhilu Zhang and Mert R. Sabuncu. "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels". In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018, pp. 8792–8802.

[15] Xiong Zhou et al. "Asymmetric Loss Functions for Learning with Noisy Labels". In: *Proceedings of the 38th International Conference on Machine Learning*. 2021, pp. 12846–12856.