ClonalNet: Classifying Better by Focusing on Confusing Categories

Xue Zhang\textsuperscript{a}, Hui-Liang Shen\textsuperscript{a,∗}

\textsuperscript{*}College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

Abstract

Existing neural classification networks predominately adopt one-hot encoding due to its simplicity in representing categorical data. However, the one-hot representation neglects inter-category correlations, which may result in poor generalization. Herein, we observe that a pre-trained baseline network has paid attention to the target image region even though it incorrectly predicts the image, revealing which categories confuse the baseline. This observation motivates us to consider inter-category correlations. Therefore, we propose a clonal network, named ClonalNet, which learns to discriminate between confusing categories derived from the pre-trained baseline. The ClonalNet architecture can be identical or smaller than the baseline architecture. When identical, ClonalNet is a clonal version of the baseline but does not share weights. When smaller, the training process of ClonalNet resembles that of the standard knowledge distillation. The difference from knowledge distillation is that we design a focusing-picking loss to optimize ClonalNet. This novel loss enforces ClonalNet to concentrate on confusing categories and make more confident predictions on ground-truth labels with the baseline reference. Experiments show that ClonalNet significantly

\textsuperscript{*}Corresponding author

Email addresses: zxue2019@zju.edu.cn (Xue Zhang), shenhl@zju.edu.cn (Hui-Liang Shen)
outperforms baseline networks and knowledge distillation.

Keywords: Image Classification, Supervised Learning, Confusing Categories, Attention Mechanism, Weighted Cross-entropy.

1. Introduction

Image classification is a crucial research field in computer vision. It forms the basis of many visual tasks, including localization [1, 2], detection [3, 4], and segmentation [2, 5]. In recent years, deep convolutional neural networks have become a leading technique for image classification tasks. Existing neural classification networks commonly represent categorical data as one-hot labels due to their simple implementation. However, much research effort has demonstrated that one-hot encoding neglects inter-category correlations [4, 6, 7, 8] as its distribution is highly discrete. For example, Szegedy et al. [6] believe that training a network with one-hot labels may result in over-fitting and produce over-confident predictions. To solve this problem, they introduce label smoothing, a weighted average of a one-hot label and a uniform distribution over categories, which improves the generalizability of networks. In addition, Hinton et al. [7] introduce soft targets in knowledge distillation. They suggest soft targets provide much more information among categories than one-hot labels, which improves the accuracy of a student model. However, enforcing networks to focus on confusing categories instead of just considering arbitrary inter-category correlations remains challenging.

To better illustrate the challenge caused by categorical confusion, in the following, we conduct an exploratory analysis using a baseline network (ResNet-18 [9]) on Imagewoof [10], a subset of ten dog breeds from ImageNet. We optimize the baseline by minimizing the cross-entropy between softmax probability
Figure 1: A schematic illustration of the key idea in this work. The baseline subpath in (a) is a pre-trained baseline network employing the softmax cross-entropy. (b) shows the softmax probabilities of the baseline for the input and two CAMs for the top-2 predictions (color borders indicate links). The baseline predicts incorrectly but concentrates on the target image region. The ClonalNet architecture in (a) is identical to or different from the baseline architecture. We optimize ClonalNet by our focusing-picking loss to discriminate between confusing categories. (c) shows the softmax probabilities and CAMs of ClonalNet that predicts correctly and pays primary attention to the target region. Here, architectures are both ResNet-18, the dataset is Imagewoof.

The pre-trained baseline makes an incorrect prediction for the input as it scores the highest on Ridgeback, but the correct label is Dingo. This failure attributes to similar appearances between Ridgeback and Dingo (readers are encouraged to compare them on the Internet). Despite this, we can observe from the CAM with a red border in Fig. (b) that the pre-trained baseline still pays attention to the target image region, i.e., the body of Dingo. Such an intriguing observation reflects which categories confuse the pre-trained baseline and motivates us...
to consider inter-category correlations.

In this work, instead of considering arbitrary correlations among categories, we intend to enable a network to focus on confusing categories determined by the pre-trained baseline. To this end, we propose a so-called ClonalNet mechanism and a novel loss function named focusing-picking loss for image classification. The ClonalNet subpath in Fig. 1 (a) illustrates the training process of ClonalNet. The image is simultaneously fed into ClonalNet and the fixed pre-trained baseline. We regard baseline logits as metrics measuring similarities between the image and different categories, based on which we infer confusing categories for each training example. The ClonalNet architecture can be either identical to the baseline architecture or smaller than the baseline architecture. When identical, ClonalNet is a clonal counterpart of the baseline but is trained using the novel loss. Thus, ClonalNet does not share weights with the pre-trained baseline. When smaller, the training process of ClonalNet is similar to that of the standard knowledge distillation [7], a technique designed to facilitate the deployment of large models. However, the significant difference from knowledge distillation is that we train ClonalNet using the novel loss. Our focusing-picking loss encourages ClonalNet to focus on confusing categories and pick a correct category. This is implemented by producing higher probabilities over confusing categories and assigning high-weighting losses to misclassified examples. In this way, Fig. 1 (c) presents the softmax probability distribution of ClonalNet and the CAMs for top-2 predictions, i.e., Dingo and Ridgeback. ClonalNet predicts correctly and pays primary attention to the target image region shown in the CAM with a red border. Our network architecture imposes no extra deployment burden on storage resources since we only deploy the ClonalNet subpath in the
inference phase. The effectiveness of the clonal mechanism has been validated on a number of baseline networks and benchmark datasets.

To summarize, the contributions of this work are mainly threefold:

• We propose a novel network mechanism, namely ClonalNet, that focuses on categories confusing the pre-trained baseline. The ClonalNet can employ architectures identical to or different from the baselines, providing alternatives to practical applications.

• We propose a focusing-picking loss for network training. Compared with the softmax cross-entropy, our loss enables ClonalNet to focus on confusing categories and assign high-weighting losses to misclassified examples.

• We conduct extensive experiments on nine challenging datasets (including two difficult fine-grained dog and bird species datasets) using ten baseline architectures. Experimental results validate that ClonalNet achieves superior performance over the baseline networks and knowledge distillation, including state-of-the-art techniques.

2. Related Work

In this section, we provide a brief overview of the deep learning methodology for improving classification performance. We roughly divide these into two categories: network architecture designing and loss function designing.

2.1. Network Architecture Designing

Designing advanced network architectures is crucial for better classification performance. Due to the success of ImageNet, the research community proposes
many classification network architectures [6, 7, 9, 11, 12], promoting wide applications of deep learning. Recently, learning inter-category correlations and adopting attention mechanisms have shown advantages in network architecture designing, such as confusing example classification [13], fine-grained category recognition [14, 15], and knowledge transfer [7, 16, 17, 18, 19, 20, 21, 22, 23].

Learning inter-category correlations is a practical strategy for confusing example classification and knowledge transfer. Classifying confusing examples refers to making correct predictions for ambiguous instances. One-hot encoding is a wide choice to represent categorical data. However, it fails to encode information among categories, leading to the risk of poor generalization [4, 8]. Sun et al. [24] propose a label embedding network architecture that generates a soft distribution for each training example, building a continuous interaction between ground-truth and other categories. Lee et al. [13] propose DropMax to drop non-target categories when calculating categorical probabilities. This architecture introduces two new layers to learn category retain probabilities and regularize the network during training. Recently, Liu et al. [25] and Guo et al. [26] propose to learn semantic correlations between sentences and corresponding ground-truth categories to integrate inter-category information in the natural language processing classification task. Transferring knowledge from a large model to a smaller one facilitates deployment. A successful case is knowledge distillation [7]. Typically, knowledge distillation requires a teacher-student model where it first trains a large teacher model and then trains a small student model by learning both one-hot labels and soft targets. A soft target reflects similar information of categories revealing how the teacher model tends to generalize.
Attention mechanism also plays a significant role in network architecture designing. This technique shows effectiveness in fine-grained category recognition as it pays more attention to discriminative image regions. For example, Fu et al. [14] propose a recurrent attention network architecture that detects discriminative regions and learns region-based representation in a mutually reinforced way. Recently, Zheng et al. [15] propose a progressive-attention network architecture to localize subtle yet discriminative parts at multiple scales progressively.

The main difference between these approaches and our work is that ClonalNet establishes links between inter-category correlations and attention mechanisms. Furthermore, we regularize ClonalNet with the attention of the pre-trained baseline, which allows ClonalNet to concentrate more on inter-category correlations and thus more on confusing categories.

2.2. Loss Function Designing

In general, deep neural networks are trained through an optimization process that minimizes a loss function. The softmax cross-entropy loss [27] is most used in multi-category classification tasks. Minimizing the cross-entropy loss can be interpreted as minimizing Kullback-Leibler divergence between empirical distributions defined by the training set and model distributions. Kullback-Leibler divergence measures the degree of dissimilarity between the two distributions [28]. The deep learning community has proposed many variants of the softmax cross-entropy loss to solve specific issues. Lin et al. [3] present a focal loss to address class imbalance problems by adding a tunable weighting factor to the cross-entropy loss. Adjusting this weighting factor allows networks to reduce relative losses for well-classified examples and focus more on hard and
Lee et al. [13] propose a softmax variant to drop non-target categories by computing a category retain probability distribution for each input instance. This variant allows classifiers to concentrate more on confusing categories. Hinton et al. [7] propose soft targets to represent the structural similarity among classes by raising the temperature of softmax, which also proves to be a compelling approach for regularizing neural networks. Szegedy et al. [6] propose label smoothing regularization, a weighted average of a one-hot label and a fixed uniform distribution over categories, which prevents networks from generating over-confident distributions among categories. Label smoothing has been used in tasks such as image classification [29], speech recognition [30], and language translation [11]. Müller et al. [31] demonstrate that the label smoothing regularization encourages intra-category compactness and equidistant inter-category separation. Yuan et al. [16] demonstrate that label smoothing regularization is a particular case of knowledge distillation while replacing the fixed uniform distribution with a learnable distribution. They introduce a self-training approach to train student models without teacher models using standard distillation loss.

Unlike the above approaches, our loss enables ClonalNet to assign higher losses to relatively hard examples by comparing pairwise probabilities between the pre-trained baseline network and ClonalNet on ground-truth categories. Furthermore, we transform the implicit attention of the baseline into an explicit regularization term which allows ClonalNet to consider confusing categories of each instance more often than others.
3. Proposed Model

In this section, we first define the notation, then introduce the ClonalNet architecture, followed by a description of how a pre-trained baseline can infer confusing categories for each training example. Finally, we show our proposed focusing-picking loss.

We denote the training set as \( D = \{(x_i, y_i)\}_{i=1}^M \) that has \( M \) training examples \( (x_i, y_i) \). \( x_i \in \mathbb{R}^d \) is the \( i \)-th input instance, and \( y_i \in \{0, 1\}^N \) is the matching one-hot label, where \( N \) is the number of categories. We aim to learn a mapping \( f : \mathbb{R}^d \to \{0, 1\}^N \) on \( D \) such that the resulting mapping \( f \) allows making a precise prediction for a test instance \( x' \) \cite{28}. For presentation convenience, we omit the subscript \( i \) in the case of a single training example.

3.1. ClonalNet Mechanism

As mentioned above, the ClonalNet mechanism can employ various combinations of network architectures and exhibits consistent performance improvements on different datasets. For illustration, Fig. 2 shows an example pipeline.

Figure 2: Example network architectures adopting ResNet-18. (a) The training process of ClonalNet. The upper subpath is the pre-trained baseline ResNet-18 employing the softmax cross-entropy. The bottom subpath is ClonalNet ResNet-18 trained by our loss. (b) The inference process of ClonalNet. It adopts the identical architecture with the baseline, without additional parameter deployment.
that adopts ResNet-18 [9] as the baseline and ClonalNet architectures, trained and tested on the Imagewoof dataset [10].

3.1.1. Training and Inference of a Baseline Network

It is known that a pre-trained network can produce a scope of categories to which each training example may belong. Due to the possible misclassification of confusing categories, the top-$k$ accuracy is consistently higher than the top-1 accuracy. This inspires us to deduce the scope of categories using a pre-trained baseline network (ResNet-18 in Fig. 2 (a)). A training example flows forward into residual convolution layers and a global average pooling layer in sequence computing a vector $h^b$ in the penultimate layer, then a 10-way fully-connected layer generating logits $z^b$, and eventually the softmax layer producing probabilities $\hat{y}^b$. The process can be formulated as

$$z^b = (W^b)^T h^b,$$

$$\hat{y}^b_n = \frac{\exp(z^b_n)}{\sum_{j=1}^{N} \exp(z^b_j)},$$

where $W^b$ is a matrix denoting weights and biases in the fully-connected layer of the baseline, $h^b$ denotes a vector including the activation in the penultimate layer of the baseline with “1” concatenated to denote a bias, $z^b_n$ and $\hat{y}^b_n$ refer to the $n$-th entries of vectors $z^b$ and $\hat{y}^b$ of the baseline, respectively.

By computing the cross-entropy between the one-hot label $y$ and softmax
where $y_n$ denotes the $n$-th entry of $y$, we can train the baseline ResNet-18 using backward propagation.

We note that the pre-trained ResNet-18 is fixed during training ClonalNet.

### 3.1.2. Training and Inference of ClonalNet

After training the baseline network, we train ClonalNet on the same dataset by minimizing our proposed focusing-picking loss. As shown in Fig. 2 (a), we feed the training example into both the pre-trained baseline and ClonalNet at the same time. These two networks compute different logits and probabilities for the training example through a similar feedforward process, but they do not share weights. It is a novel operation in the baseline subpath when training ClonalNet that inferring a multi-warm label from baseline logits. Our proposed multi-warm label is a mathematical representation of confusing categories, which is designed based on the observation that logits can be regarded as a set of similarity metrics between an image and different categories. Our proposed focusing-picking loss optimizes ClonalNet by taking re-weighted probabilities, softmax probabilities, a multi-warm label, and a one-hot label as inputs. Parameters in the pre-trained baseline are not updated. We will introduce more details about our loss designing in Section 3.3.

As shown in Fig. 2 (b), only ClonalNet is needed for inference and deployment, and thus the ClonalNet architecture does not require additional parame-
ters compared to the baseline.

### 3.2. Multi-warm Label

To guide the attention of ClonalNet, we employ the pre-trained baseline to generate confusing categories, mathematically represented as a multi-warm label, for each training example. Specifically, the pre-trained baseline generates logits $z^b$ using (1). A logit $z^b_n$ is a dot product given an activation vector $h^b$ and a weighting vector $w^b_n$, with which we can compute the cosine similarity between the two vectors as

$$S(w^b_n, h^b) = \frac{(w^b_n)^\top h^b}{||w^b_n|| \cdot ||h^b||},$$  

where $w^b_n$ is a weighting vector of matrix $W^b$ corresponding to the $n$-th category, and $||\cdot||$ denotes the magnitude of a vector. The weighting vector $w^b_n$ serves as the template of a category and keeps constant [31]. In this regard, a positive or negative cosine similarity value indicates that these two vectors point in close or opposite directions, while a zero value indicates that they are orthogonal.

We propose a multi-warm label vector whose $n$-th entry is determined as

$$l'_n = \begin{cases} 
1, & \text{if } S(w^b_n, h^b) > 0 \\
0, & \text{others}
\end{cases}$$

Then, we add the ground-truth label of the training example into the multi-warm label and then clip the resulting multi-warm label for learning stability [13],

$$l''_n = \min(1, l'_n + y_n).$$
Figure 3: The relationship between the attention of the pre-trained baseline and our proposed multi-warm label. The pre-trained baseline process the input image in each scene to generate CAMs, logits, and the multi-warm label. CAMs highlight discriminative regions for corresponding categories. The positive and negative logits are marked in red and gray, respectively. Non-zero entries of the multi-warm label and the CAMs of confusing categories are marked in red bars and red borders, respectively. The “Category” row presents categorical indexes and categorical names.

We further normalize the multi-warm label as

$$l_n = \frac{l_n'}{\sum_{j=1}^{N} l_j'}.$$  \hspace{1cm} (7)  

Then the multi-warm label $l = (l_1, l_2, \ldots, l_N)^T$. The non-zero entries of the label vector $l$ are of the same probability and sum up to 1.0. Generating the multi-warm label is a straightforward process that does not require learning additional parameters and, therefore, does not increase the deployment burden of neural networks.

A multi-warm label reflects the attention of the pre-trained baseline. In other words, it transforms the implicit attention of the pre-trained baseline into explicit knowledge, which can be learned by ClonalNet directly. Specifically, we show CAMs of various categories, corresponding logits, and multi-warm labels in Fig. 3. For Scene-A, the ground-truth is Beagle. The baseline focuses on Beagle (index: 2), Samoyed (index: 8), and Dingo (index: 9). It generates positive logits on these categories as they are similar in appearance (readers are
encouraged to compare them on the Internet), which implies that the multi-warm label reflects the attention of the baseline. Both Scene-B and Scene-C contain Dingoes, identical in species but vary in fur colors (yellow and white, respectively). The baseline generates significantly different CAMs and multi-warm labels, demonstrating that each input’s multi-warm label is adaptive. In addition, compared to one-hot labels, multi-warm labels are characterized by higher entropy and generalize well [8].

3.3. Loss Function

ClonalNet converts logits $z$ into predicted probabilities $\hat{y}$ as

$$\hat{y}_n = \frac{\exp(d_n + z_n)}{\sum_{j=1}^{N} \exp(d_j + z_j)},$$

(8)

where

$$d_n = \frac{\exp(z_n)}{\sum_{j=1}^{N} \exp(z_j)} - \frac{\exp(z_n^b)}{\sum_{j=1}^{N} \exp(z_j^b)},$$

(9)

where $z_n$ and $\hat{y}_n$ are the logit and predicted probability on the $n$-th category in ClonalNet, respectively. $d_n$ is the difference between pairwise softmax probabilities of ClonalNet and the pre-trained baseline. It is the entry of differences $d$ at index $n$. The normal softmax probability distribution in [2] is a particular case of the re-weighted probability distribution in [8] when $d = 0$. We denote the normal and re-weighted probability distributions of ClonalNet for notation simplicity as $\hat{y}(z, d = 0)$ and $\hat{y}(z, d)$, respectively. A positive difference on the ground-truth category implies that the input is a relatively easy example since ClonalNet generates a more confident prediction than the baseline. In contrast,
a negative difference corresponds to a relatively hard or misclassified example. The role of differences is to force ClonalNet to tune weights for logits so that relatively easy and hard examples produce a higher and lower probability on the ground-truth category, respectively.

We compute the cross-entropy between re-weighted probabilities and a one-hot label as classification loss

$$L_{\text{cls}} = H(y, \hat{y}(z, d))$$

$$= - \sum_{n=1}^{N} y_n \log (\hat{y}_n(z, d))$$

$$= - \log \left( \frac{\exp(d_t + z_t)}{\sum_{j=1}^{N} \exp(d_j + z_j)} \right),$$

where index $t$ refers to the ground-truth category. Such a design allows ClonalNet to take the prediction from the pre-trained baseline as the reference and assigns greater losses to relatively hard examples. At testing time, ClonalNet uses differences of $d = 0$, i.e., the normal softmax.

To guide the attention of ClonalNet to confusing categories during training, we propose to regularize the network by minimizing the cross-entropy between normal softmax probabilities and multi-warm labels

$$R_{\text{attention}} = H(l, \hat{y}(z, d = 0))$$

$$= - \sum_{n=1}^{N} l_n \log (\hat{y}_n(z, d = 0))$$

$$= - \sum_{n=1}^{N} l_n \log \left( \frac{\exp(z_n)}{\sum_{j=1}^{N} \exp(z_j)} \right).$$

This regularization encourages ClonalNet to generate high probabilities on confusing categories and penalizes the others. For example, Scene-B in Fig. 3 shows
that the input Dingo (index: 9) confuses the pre-trained baseline by indexes of 1, 2, 4, and 9, respectively. ClonalNet should focus on them. Regularized by this a priori knowledge using (11), ClonalNet generates higher probabilities on these four categories than the others, as shown in the ClonalNet subpath of Fig. 1 (c).

Last but not least, minimizing the classification loss in (10) encourages ClonalNet to make more confident predictions on ground-truth categories, which leads it to, more likely, produce over-confident distributions, i.e., low-entropy distributions. Preventing this potential degradation is critical for ClonalNet to generalize well on test sets. To this end, we penalize low-entropy distributions by maximizing the entropy of normal softmax probabilities. We compute the entropy of ClonalNet for each training example as

\[
R_{\text{entropy}} = H(\hat{y}(z, d = 0)) \\
= -\sum_{n=1}^{N} \hat{y}_n(z, d = 0) \log(\hat{y}_n(z, d = 0)) \\
= -\sum_{n=1}^{N} \frac{\exp(z_n)}{\sum_{j=1}^{N} \exp(z_j)} \log \left( \frac{\exp(z_n)}{\sum_{j=1}^{N} \exp(z_j)} \right).
\]

Adding negative entropy to the classification loss allows ClonalNet to produce smoother probability distributions with higher entropy and thus generalize better, similar to label smoothing.

To train ClonalNet in Fig. 2 (a), we propose a focusing-picking loss consisting of the three components described above, which aims to encourage ClonalNet to perform a high weight loss on relatively hard training examples, focus on confusing categories, and take into account the prevention of low-entropy output.
distributions, which is defined as

\[ L = L_{\text{cls}} + \alpha R_{\text{attention}} - \beta R_{\text{entropy}} \]

\[ = H(y, \hat{y}(z, d)) + \alpha H(1, \hat{y}(z, d = 0)) - \beta H(\hat{y}(z, d = 0)). \] (13)

We experimentally find that adopting \( \alpha = 0.1, \beta = 1.0 \) generally exhibits favorable performance improvement.

4. Experiments

In this section, we evaluate the performance of ClonalNet on object classification and material recognition tasks.

On the object classification task, we compare ClonalNet with various baseline networks, DropMax [13], standard knowledge distillation [7], self-training knowledge distillation [16], and self-knowledge distillation [17, 18, 19, 20, 21, 22, 23]. DropMax [13] attempts to make the neural network more focused on confusing categories. Its motivation is similar to our ClonalNet. Standard and self-training knowledge distillation [7, 16] use pre-trained teacher models to provide soft targets to the classification network. Standard knowledge distillation generally requires the pre-trained teacher model is larger and stronger, while self-training knowledge distillation allows the pre-trained model to be itself. ClonalNet is similar to the two techniques in terms of the training strategy. It also needs a pre-trained baseline network that provides multi-warm labels. The network architecture of ClonalNet can be identical with or different from the baseline architecture. Self-knowledge distillation [17, 18, 19, 20, 21, 22, 23] does not require training a stronger and larger teacher model in the training phase. It does not increase the deployment burden compared to the baseline architecture.
in the testing phase, structurally similar to ClonalNet. On the material recognition task, we evaluate ClonalNet’s classification performance using various baseline architectures on three material datasets. We also compare ClonalNet with state-of-the-art approaches on both object classification and material recognition tasks.

4.1. Datasets

We introduce the public datasets used in our experiments, including six object classification datasets and three material recognition datasets. Table 1 lists the detailed statistics with the number of categories and data splits.

MNIST (Modified National Institute of Standards and Technology Dataset) [32] is a famous dataset of handwritten digits from 0 to 9. It consists of 10 categories, with 60,000 training examples and 10,000 test examples.

CIFAR-10 and CIFAR-100 (Canadian Institute for Advanced Research Dataset) [33] both contain 60,000 examples, 50,000 for training and 10,000 for testing. The difference is that the former is organized into 10 categories while the latter is organized into 100 categories.

Imagewoof [10] is a subset of ImageNet. This dataset contains 10 dog breeds.
that are not easy to classify. It contains 9,025 training examples and 3,929 test examples.

CUB-200-2011 (Caltech-UCSD Birds 200-2011 Dataset) [34] is a widely-used fine-grained bird species dataset for object classification. It contains 11,788 examples with 200 bird species, 5,994 for training and 5,794 for testing.

Stanford Dogs [35] is a fine-grained dog species dataset. It contains 120 dog breeds and 20,580 examples, out of which 12,000 are used for training and 8,580 for testing.

MINC-2500 (Material in Context Dataset-2500) [5] is used to recognize material in real-world photos. It consists of 23 categories of everyday materials in the wild, and each category evenly includes 2,500 examples, with 2,125 for training, 125 for validation, and 250 for testing.

DTD (Describable Textures Dataset) [36] is a texture dataset. It consists of 47 categories with 120 examples per category, 40 for training, 40 for validation, and 40 for testing.

GTOS-mobile (An Extended Version of Ground Terrain in Outdoor Scenes Dataset) [37] is used to recognize outdoor terrain. It consists of 31 categories containing a total of 31,315 training examples and 6,066 test examples. All test instances are taken with a mobile phone.

4.2. Object Classification

We first investigate the effectiveness of ClonalNet for general classification tasks. Then, we show that the proposed focusing-picking loss can transfer knowledge from large network architecture to small network architecture. We also perform comprehensive comparisons with related algorithms.
4.2.1. Comparison with Baseline and DropMax

For a fair comparison, we generally combine networks with datasets following the practice in DropMax [13]. We train LeNet [32] on MNIST-55K, train ResNet-20 [9] on CIFAR-10, train ResNet-34 [9] on CIFAR-100, and train ResNet-18 [9] on CUB-200-2011. The MNIST-55K dataset has 55K, 5K, and 10K training, validation, and testing instances. We select the fine-grained dataset CUB-200-2011, which contains many ambiguous examples, to validate neural networks can improve performances by focusing on confusing categories.

We train a baseline using the normal softmax cross-entropy in (3). The resulting pre-trained baseline provides the multi-warm label for each training example employing (7). We train ClonalNet with the identical architecture to the baseline by minimizing our focusing-picking loss in (13) and deploy the resulting well-trained ClonalNet for inference. We compare ClonalNet with DropMax [13] since they both enforce networks to focus on confusing categories. Their difference in network architecture is that DropMax needs to deploy an additional fully connected layer to infer the class retain probability for each test instance. However, ClonalNet does not change the baseline architecture.

Table 2 shows top-1 test accuracy. From the results, we can make three observations. (1) DropMax shows consistent improvements over the baselines across all datasets, and it obtains a noticeably relative gain on CUB-200-2011. (2) ClonalNet achieves the best accuracy and significantly outperforms Drop-
Table 3: The number of trainable variables for various neural networks used in Table 4, where \( K \) and \( N \) refer to the number of rows and columns of the fully-connected matrix, respectively.

| Neural Network | Input Size | Parameters |
|----------------|------------|------------|
| ResNet-20 [9]  | 32 × 32 × 3 | \( N(K + 1) \times (K + 1) \) |
| WRN-28-10 [38] | 36 × 36 × 3 | \( N(K + 1) \times (K + 1) \) |
| ResNet-18 [9]  | 224 × 224 × 3 | \( N(K + 1) \times (K + 1) \) |
| ResNet-34 [9]  | 224 × 224 × 3 | \( N(K + 1) \times (K + 1) \) |
| ResNet-50 [9]  | 224 × 224 × 3 | \( N(K + 1) \times (K + 1) \) |
| InceptionV3 [6]| 299 × 299 × 3 | \( N(K + 1) \times (K + 1) \) |

Table 4: Comparisons of top-1 test accuracy with knowledge distillation on CIFAR-100 and CUB-200-2011 over five runs. The “Individual Training” column shows the results of individual networks using the normal softmax cross-entropy. “Knowledge Distillation” and “ClonalNet” columns are the results of Net-2 using the standard distillation loss and the proposed focusing-picking loss, respectively.

| Networks          | Individual Training | Accuracy of Net-2 Using Different Loss Functions |
|-------------------|---------------------|-------------------------------------------------|
|                   | Net-1               | Net-2                                           |
| CIFAR-100         | 72.56% ± 0.13%      | 66.81% ± 0.22%                                 |
| WRN-28-10         | 77.98% ± 0.20%      | 67.25% ± 0.40% (↑ 0.44%)                       |
| CUB-200-2011      | 77.43% ± 0.54%      | 62.10% ± 0.10% (↑ 10.94%)                      |
| InceptionV3       | 81.31% ± 0.24%      | 64.65% ± 0.14% (↑ 10.50%)                      |
| ResNet-34         | 58.63% ± 0.26%      | 59.42% ± 0.24% (↑ 0.79%)                       |

Max on each dataset. For example, ClonalNet obtains the relative improvement of 5.64% over the baseline on CIFAR-100, much higher than DropMax with 0.73% relative improvement. (3) ClonalNet can effectively classify ambiguous instances. For example, ClonalNet boosts the accuracy on CUB-200-2011 to 62.21% with 11.05% relative gains compared with the baseline, remarkably outperforming DropMax with 7.77% relative gains. The significant improvement of ClonalNet attributes to the pre-trained baseline that provides the multi-warm label with rich inter-category correlations for each training example, with which ClonalNet learns to focus on the confusing categories and discriminate the correct category.

The above experiments demonstrate that ClonalNet can successfully improve the performance of the baseline network by focusing on confusing categories, given the same architecture as the baseline network.

4.2.2. Comparison with Knowledge Distillation

When using a smaller network architecture than the baseline architecture, ClonalNet’s training process is similar to the standard knowledge distillation [7].
However, they draw inspiration from a distinct aspect. The teacher-student network in [7] is designed to distill knowledge from a large teacher model to a small student model. ClonalNet is intended to discriminate the correct category from the confusing ones and does not require training a stronger teacher model first.

We compare classification performances of seven modern networks on two representative datasets, CIFAR-100 and CUB-200-2011, and list the number of trainable parameters of each network in Table 3. Table 4 shows the top-1 test accuracy of each network, knowledge distillation, and ClonalNet on both datasets. In the first four combinations in the “Networks” column, we choose large models as Net-1 and small models as Net-2. In the last combination, we choose ResNet-34 as Net-1 and Net-2, with the training process similar to the self-training knowledge distillation in [16]. From Table 4, we can observe the following four phenomena. (1) For all combinations of networks, ClonalNet consistently performs the best. It significantly outperforms Net-2 trained individually and knowledge distillation, which we can visualize by comparing the relative improvements in parentheses. (2) When Net-1 and Net-2 are ResNet-34, knowledge distillation obtains a slight improvement (0.79% relative gain) over the baseline. However, ClonalNet obtains a significant improvement with a relative gain of 5.02%. (3) The performances of ClonalNet and knowledge distillation does not depend entirely on the capacity of Net-1. In our experiments, both ResNet-20 and ResNet-18 tend to perform better when combined with residual networks. (4) ClonalNet-ResNet-18 significantly outperforms Individual-Training-ResNet-34 and Knowledge-Distillation-ResNet-34 on CUB-200-2011, even though ResNet-34 is twice as deep as ResNet-18.
The above experiments show that our focusing-picking loss is practical for ClonalNet with various architectures. More importantly, ClonalNet generally outperforms standard knowledge distillation and self-training knowledge distillation, suggesting that it can be a reasonable choice for improving the performance of small models when learning from large teacher models or when stronger teacher models are not available.

4.2.3. Comparison with Self-Knowledge Distillation

Both ClonalNet and self-knowledge distillation techniques do not require training larger and stronger teacher models in the training phase. Neither of them adds computational and deployment burden to the baseline network architecture in the test phase, so we compare their classification performance.

The designing philosophies and training strategies of the two techniques are different. Self-knowledge distillation distills own deep or refined features with valuable knowledge to its shallow layers. In the training phase, self-knowledge distillation generally requires training a series of additional auxiliary networks along with the classification network. ClonalNet focuses more on discriminating the correct category from the confusing categories of each example. In the training phase, ClonalNet only trains the classification network without training any other auxiliary networks, which keeps the trainable parameters of ClonalNet the same as the corresponding baseline network architecture.

We compare ClonalNet with seven self-knowledge distillation methods. Self-knowledge distillation commonly includes data augmentation-based methods such as DDGSD [18], CS-KD [21], SLA-SD [22], and techniques using auxiliary networks such as BYOT [19], ONE [17], SAD [20], FRSKD [23]. Among these methods, FRSKD shows state-of-the-art performance in image classifica-
tion tasks on various datasets. It uses an auxiliary teacher to provide refined feature maps and soft targets to the classification network. The loss function of FRSKD is the weighted average of cross-entropy, standard distillation loss, and feature distillation loss, which is denoted as $L_{\text{FRSKD}} = L_{\text{CE}}^c + L_{\text{CE}}^t + \mu L_{\text{KD}} + \nu L_F$, where $L_{\text{CE}}^c$ and $L_{\text{CE}}^t$ are the cross-entropy loss functions of classification network and self-teacher network, $L_{\text{KD}}$ and $L_F$ are the standard distillation loss with soft targets and the feature distillation loss with refined feature maps. In addition, FRSKD\textbackslash F \cite{23} does not use feature distillation but only soft targets distillation. Its loss function is $L_{\text{FRSKD}\textbackslash F} = L_{\text{CE}}^c + L_{\text{CE}}^t + \mu L_{\text{KD}}$.

ClonalNet does not constrain the network architecture’s cross-layer features and intermediate feature layers but learns to focus on confusing categories in the output layer, structurally similar to FRSKD\textbackslash F. For a fair comparison, we compare ClonalNet with FRSKD\textbackslash F.

Our proposed focusing-picking loss is one of our main contributions, which allows the neural network to focus on confusing categories and assign higher losses to misclassified examples. It can be used as a general loss function in a network architecture similar to ClonalNet. We replace the cross-entropy of classification network and soft targets distillation loss in the loss function of FRSKD with the proposed focusing-picking loss. We denote this experimental setting as OurLoss+FRSKD, which is specifically $L_{\text{OurLoss+FRSKD}} = L + L_{\text{CE}}^t + \nu L_F$, where $L$ is our proposed focusing-picking loss in \cite{13}.

We use ResNet-18 as the baseline network architecture on three datasets, including CIFAR-100 and two fine-grained datasets, CUB-200-2011 and Stanford Dogs, to conduct comparison experiments. To adapt ResNet-18 to CIFAR-100 with a small input size, we modify the first convolutional layer to a $3\times3$ convol-
Table 5: Performance comparison on CUB-200-2011, Stanford Dogs, and CIFAR-100. The baseline architecture is ResNet-18. The results are the average and standard deviation of accuracy over three runs. The best accuracy on each dataset is bolded. Underlines mark the best accuracy between FRSKD\F and ClonalNet.

| Methods       | CUB-200-2011       | Stanford Dogs     | CIFAR-100        |
|---------------|--------------------|--------------------|------------------|
| Baseline      | 51.72% ± 1.17%     | 63.38% ± 0.04%     | 73.80% ± 0.60%   |
| ONE           | 54.71% ± 0.42%     | 65.39% ± 0.59%     | 76.67% ± 0.66%   |
| DDGSD         | 58.49% ± 0.55%     | 69.00% ± 0.28%     | 76.61% ± 0.47%   |
| BYOT          | 58.66% ± 0.51%     | 68.82% ± 0.15%     | 76.68% ± 0.07%   |
| SAD           | 52.76% ± 0.57%     | 63.17% ± 0.56%     | 74.65% ± 0.33%   |
| CS-KD         | 64.34% ± 0.08%     | 68.91% ± 0.40%     | 77.19% ± 0.05%   |
| SLA-SD        | 56.17% ± 0.71%     | 67.30% ± 0.21%     | 77.52% ± 0.30%   |
| FRSKD         | 65.39% ± 0.13%     | 70.77% ± 0.20%     | 77.71% ± 0.14%   |
| FRSKD\F       | 62.29% ± 1.65%     | 69.48% ± 0.84%     | 77.64% ± 0.12%   |
| ClonalNet(Ours)| 62.77% ± 0.05%    | 69.81% ± 0.47%     | 77.43% ± 0.56%   |
| OurLoss+FRSKD | 67.21% ± 0.17%     | 71.50% ± 0.21%     | 76.89% ± 0.30%   |

We also remove the max-pooling layer. For the other two fine-grained datasets, we use the standard ResNet-18. Table 5 lists the classification performance. Observing the results, we can find the following phenomena. (1) Among all the self-knowledge distillation methods, including ONE, DDGSD, BYOT, SAD, CS-KD, SLA-SD, and FRSKD, FRSKD achieves the best performance. (2) ClonalNet outperforms several self-knowledge distillation techniques, and this advantage is more notable on fine-grained datasets due to its greater focus on confusing categories. For example, ClonalNet significantly outperforms ONE, DDGSD, BYOT, SAD, SLA-SD on the CUB-200-2011 dataset. (3) ClonalNet and FRSKD\F perform comparably in average accuracy when there are no constrained intermediate layer features, e.g., ClonalNet achieves 62.77% and 69.81% average accuracy on the fine-grained datasets CUB200-2011 and Stanford Dogs, respectively, slightly higher than FRSKD\F. (4) OurLoss+FRSKD consistently achieves the best performance on CUB200-2011 and Stanford Dogs, outperforming the state-of-the-art FRSKD by a relative improvement of 1.82% and 0.73% in average accuracy,
respectively, which illustrates the success of our focusing-picking loss in classifying confusing categories. On CIFAR-100, FRSKD maintains the lead.

4.2.4. Implementation Details for Object Classification

In sections 4.2.1 and 4.2.2, each baseline network and its corresponding DropMax, knowledge distillation, and ClonalNet are trained with the same data augmentation methods and optimization strategies. When training and testing LeNet on MNIST-55K, we use the official implementation in [13]. For experiments on CIFAR-10 and CIFAR-100, we follow He’s practices [9]. For experiments on CUB-200-2011, we train networks for 100 epochs with a mini-batch size of 64. We set the temperature to 2.5 and the weighting factor to 0.6, respectively, when conducting knowledge distillation. In our experiments, we normalize input images by subtracting a per-channel mean value. We optimize networks using SGD with a momentum of 0.9 during training and only evaluate original images without data augmentation during testing. In section 4.2.3, we use available official implementations and follow the same practices in [23].

4.3. Material Recognition

4.3.1. Comparison with Baseline

We conduct comprehensive experiments on three material datasets, MINC-2500, DTD, and GTOS-mobile, using three widely used network architectures, including InceptionV3 [6], ResNet-50 [9], and MobileNetV3 [12]. As in previous training procedures, we train baselines on each dataset employing the normal softmax cross-entropy. The resulting pre-trained baseline generates the multi-warm label for each training example. When training ClonalNet, we adopt
Table 6: Comparisons of top-1 test accuracy with baseline networks on MINC-2500, DTD, and GTOS-mobile over five runs.

| Baseline Network | ClonalNet (Ours) |
|------------------|------------------|
| **InceptionV3**  |                  |
| MINC-2500        | 82.22% ± 0.22%   |
|                  | 85.37% ± 0.12% (↑ 3.15%) |
| DTD              | 72.80% ± 0.58%   |
|                  | 74.97% ± 0.28% (↑ 2.17%) |
| GTOS-mobile      | 74.69% ± 3.03%   |
|                  | 83.32% ± 1.47% (↑ 8.63%) |
| **ResNet-50**    |                  |
| MINC-2500        | 77.37% ± 0.20%   |
|                  | 79.76% ± 0.27% (↑ 2.39%) |
| DTD              | 67.44% ± 0.46%   |
|                  | 72.32% ± 0.53% (↑ 4.88%) |
| GTOS-mobile      | 78.10% ± 2.11%   |
|                  | 80.03% ± 1.26% (↑ 1.93%) |
| **MobileNetV3**  |                  |
| MINC-2500        | 79.60% ± 0.26%   |
|                  | 81.06% ± 0.04% (↑ 1.46%) |
| DTD              | 71.17% ± 0.35%   |
|                  | 73.11% ± 0.22% (↑ 1.94%) |
| GTOS-mobile      | 75.56% ± 0.81%   |
|                  | 78.01% ± 0.53% (↑ 2.45%) |

Table 6 shows the classification performances. Comparing with the baseline network on each dataset, we can observe that ClonalNet consistently achieves the best performances. For example, ClonalNet-InceptionV3 and ClonalNet-ResNet-50 achieve 85.37%, 83.32%, and 72.32% accuracy on MINC-2500, GTOS-mobile, and DTD, exhibiting 3.15%, 8.63%, and 4.88% relative improvements compared with baselines. In addition, ClonalNet-InceptionV3 with larger input image size performs better than ClonalNet-ResNet-50 and ClonalNet-MobileNetV3.

### 4.3.2. Comparison with DEP

Table 7: Comparisons of top-1 test accuracy with baseline DEP on MINC-2500, DTD, and GTOS-mobile over five runs.

| Baseline Network | ClonalNet (Ours) |
|------------------|------------------|
| **DEP (the previous SOTA)** |                  |
| MINC-2500        | 81.56% ± 0.22%   |
|                  | 82.25% ± 0.14% (↑ 0.69%) |
| DTD              | 73.34% ± 0.26%   |
|                  | 73.83% ± 0.14% (↑ 0.49%) |
| GTOS-mobile      | 76.76% ± 0.96%   |
|                  | 79.57% ± 1.11% (↑ 2.81%) |

We further experiment using the state-of-the-art network DEP [37] on ma-
material datasets. The backbone architecture of DEP is ResNet-18 on GTOS-mobile and ResNet-50 on both MINC-2500 and DTD. We re-implement DEP on the three material datasets as baselines for a fair comparison and ensure that the results are comparable even outperform those reported in the original paper [37]. Afterwards, we train our ClonalNet-DEP with identical architectures to the baseline by minimizing the proposed focusing-picking loss. Table 7 lists top-1 test accuracy. From the results, we can observe that our ClonalNet-DEP improves Baseline-Network-DEP further across all datasets. For example, ClonalNet-DEP boosts the accuracy to 79.57% with 2.81% improvement relative to Baseline-Network-DEP on GTOS-mobile. In addition, ClonalNet-ResNet-50 in Table 6 with 72.32% accuracy on DTD approaches the performance of Baseline-Network-DEP with 73.34% accuracy in Table 7. This comparable performance is because of many confusing categories in the DTD dataset, such as category “lined”, category “banded”, and category “striped”. ClonalNet can better classify ambiguous examples by focusing more on these confusing categories.

These experiments show that ClonalNet can be applied to material recognition tasks and effectively improve the performance of baselines with various network architectures, which implies our clonal mechanism is a general approach to gains higher accuracy.

4.3.3. Implementation Details for Material Recognition

We use the same data augmentation methods and optimization schemes as in section 4.2.4 for each baseline network and its clonal version. We train networks on MINC-2500, DTD, and GTOS-mobile for 10, 100, and 30 epochs, determined by the number of training examples and the classification accuracy of validation sets. For DEP and ClonalNet-DEP training and testing, we follow
official implementations in [37].

5. Conclusion

This work proposes ClonalNet and the focusing-picking loss to discriminate a ground-truth label from confusing categories. We establish the link between confusing categories and the attention mechanism of the pre-trained baseline. We mathematically represent confusing categories as the proposed multi-warm label with valuable inter-category correlations based on cosine similarity. We have shown that the focusing-picking loss enables ClonalNet to focus on confusing categories and high-weight contributions of relatively hard examples with the baseline reference. Our ClonalNet mechanism is quite simple and generally effective without imposing extra deployment burdens on storage resources. We have conducted extensive experiments and presented that various baseline architectures benefit from our ClonalNet mechanism. Applications of ClonalNet are extensive. For example, a state-of-the-art self-knowledge distillation technique combined with our focusing-picking loss achieves improved performance on two fine-grained datasets. A state-of-the-art network architecture used for material recognition gains performances further when using our ClonalNet mechanism. Transferring knowledge from large models into small models using our focusing-picking loss notably outperforms the standard knowledge distillation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 81973751.
References

[1] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[2] J. Choe, S. Lee, H. Shim, Attention-based dropout layer for weakly supervised single object localization and semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017.

[4] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[5] S. Bell, P. Upchurch, N. Snavely, K. Bala, Material recognition in the wild with the materials in context database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3479–3487.

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[7] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: Advances in Neural Information Processing Systems Workshop, 2015.
[8] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, G. Hinton, Regularizing neural networks by penalizing confident output distributions, International Conference on Learning Representations Workshop (2017).

[9] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[10] J. Howard, Imagewoof. URL https://github.com/fastai/imagenette

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.

[12] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for MobileNetV3, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[13] H. B. Lee, J. Lee, S. Kim, E. Yang, S. J. Hwang, DropMax: Adaptive variational softmax, in: Advances in Neural Information Processing Systems, 2018.

[14] J. Fu, H. Zheng, T. Mei, Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[15] H. Zheng, J. Fu, Z.-J. Zha, J. Luo, T. Mei, Learning rich part hierar-
chies with progressive attention networks for fine-grained image recognition, IEEE Transactions on Image Processing 29 (2019) 476–488.

[16] L. Yuan, F. E. Tay, G. Li, T. Wang, J. Feng, Revisiting knowledge distillation via label smoothing regularization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 3903–3911.

[17] X. Lan, X. Zhu, S. Gong, Knowledge distillation by on-the-fly native ensemble, in: Advances in Neural Information Processing Systems, 2018.

[18] T.-B. Xu, C.-L. Liu, Data-distortion guided self-distillation for deep neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[19] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, K. Ma, Be your own teacher: Improve the performance of convolutional neural networks via self distillation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[20] Y. Hou, Z. Ma, C. Liu, C. C. Loy, Learning lightweight lane detection CNNs by self attention distillation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[21] S. Yun, J. Park, K. Lee, J. Shin, Regularizing class-wise predictions via self-knowledge distillation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[22] H. Lee, S. J. Hwang, J. Shin, Self-supervised label augmentation via input transformations, in: International Conference on Machine Learning, 2020.
[23] M. Ji, S. Shin, S. Hwang, G. Park, I.-C. Moon, Refine myself by teaching myself: Feature refinement via self-knowledge distillation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021.

[24] X. Sun, B. Wei, X. Ren, S. Ma, Label embedding network: Learning label representation for soft training of deep networks, arXiv preprint arXiv:1710.10393 (2017).

[25] Y. Liu, X. Liu, W. Hu, L. Zhang, M. Zhang, A label-enhanced text classification model, Journal of Physics: Conference Series (2020).

[26] B. Guo, S. Han, X. Han, H. Huang, T. Lu, Label confusion learning to enhance text classification models, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[27] E. B. Baum, F. Wilczek, Supervised learning of probability distributions by neural networks, in: Advances in Neural Information Processing Systems, 1988.

[28] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT Press Cambridge, 2016.

[29] M. Lukasik, S. Bhojanapalli, A. Menon, S. Kumar, Does label smoothing mitigate label noise?, in: International Conference on Machine Learning, 2020.

[30] M. Song, Y. Zhao, S. Wang, M. Han, Word similarity based label smoothing in RNNLM training for ASR, in: IEEE Spoken Language Technology Workshop, 2021.
[31] R. Müller, S. Kornblith, G. E. Hinton, When does label smoothing help?, in: Advances in Neural Information Processing Systems, 2019.

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

[33] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Technical Report (2009).

[34] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 dataset, Tech. rep. (2011).

[35] A. Khosla, N. Jayadevaprakash, B. Yao, F.-F. Li, Novel dataset for fine-grained image categorization: Stanford dogs, in: IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2011.

[36] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.

[37] J. Xue, H. Zhang, K. Dana, Deep texture manifold for ground terrain recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 558–567.

[38] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of the British Machine Vision Conference, 2016.