Exploitation of Noisy Automatic Data Annotation and Its Application to Hand Posture Classification

Georgios Lydakis[1,2], Iason Oikonomidis[2][a], Dimitrios Kosmopoulos[3][b] and Antonis A. Argyros[1,2][c]
[1] *Foundation for Research and Technology - Hellas (FORTH), Greece*
[2] *Computer Science Department, University of Crete, Greece*
[3] *University of Patras, Greece*

Keywords: Automatic Data Annotation, Noisy Annotation, Hand Posture Classification.

Abstract: The success of deep learning in recent years relies on the availability of large amounts of accurately annotated training data. In this work, we investigate a technique for utilizing automatically annotated data in classification problems. Using a small number of manually annotated samples, and a large set of data that feature automatically created, noisy labels, our approach trains a Convolutional Neural Network (CNN) in an iterative manner. The automatic annotations are combined with the predictions of the network in order to gradually expand the training set. In order to evaluate the performance of the proposed approach, we apply it to the problem of hand posture recognition from RGB images. We compare the results of training a CNN classifier with and without the use of our technique. Our method yields a significant increase in average classification accuracy, and also decreases the deviation in class accuracies, thus indicating the validity and the usefulness of the proposed approach.

1 INTRODUCTION

In the past few years, deep learning methods have revolutionized the field of Artificial Intelligence (AI), achieving previously unattainable performance on a plethora of challenging tasks. Examples include image recognition (He et al., 2015), natural language processing and machine translation (Cho et al., 2014) and speech recognition (Graves et al., 2013).

In the case of supervised learning, annotating large amounts of data can quickly become very costly in terms of human effort, especially so if the annotation procedure is itself difficult, for example, creating pixel-level segmentation masks. For this reason, besides semi-supervised and unsupervised learning, research is also highly active in the field of reducing the annotation effort required for training deep models in a supervised manner. In this work, we propose a technique for utilizing a large number of samples that have been automatically annotated with labels for a classification task. Given that the annotation is automatic, it is possible that the extracted labels may be noisy. Reliable automatic annotation systems are generally very hard if not impossible to design and build, depending on the targeted problem. Generally, most current Computer Vision approaches can yield somewhat reliable results under specific, controlled scenarios, but in general it is unavoidable to have failure cases, to a varying degree.

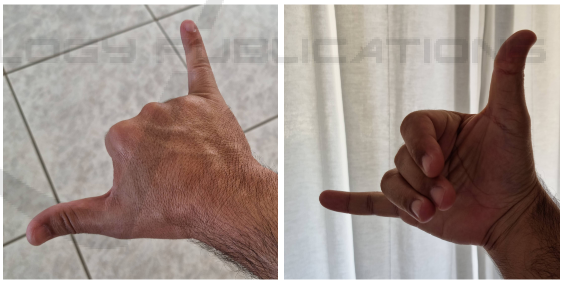

Figure 1: A classifier trained only on a small dataset may fail to recognize that the hand posture depicted in these two images is the same. We propose a method to exploit automatically annotated, noisy data in order to train a better hand posture classifier.

The method we develop is generic in nature, and can be tailored to address any classification problem. It assumes the existence of a small number of manually annotated samples, as well as a large set of au-

[a] https://orcid.org/0000-0002-9503-3723
[b] https://orcid.org/0000-0003-3325-1247
[c] https://orcid.org/0000-0001-8230-3192

Lydakis, G., Oikonomidis, I., Kosmopoulos, D. and Argyros, A.
Exploitation of Noisy Automatic Data Annotation and Its Application to Hand Posture Classification.
DOI: 10.5220/0010906200003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 632-641
ISBN: 978-989-758-555-5; ISSN: 2184-4321

tomatically annotated ones, whose labels might be noisy. We begin by training a Convolutional Neural Network (CNN) on the manually annotated data, resulting in a classifier that might not generalize well, given the small number of samples.

Then, we compare the predictions of the network for all automatically annotated samples with the noisy ground truth labels. We incorporate in the training set a subset of those, for which the predictions of the classifier agrees with the labels. The intuition is that the agreement of the two predictors (classifier, noisy automatic annotation) is probably not coincidental. The network is then trained again on the new dataset, and the procedure is iteratively repeated until there is no improvement in validation accuracy. We experiment with two variations of the method, based on how conservatively/aggressively we expand the training dataset with automatically annotated data.

In order to evaluate our technique, we apply it to the problem of hand posture recognition from RGB images (see Figure 1). The posture and motion of hands play an important role in conveying information in sign languages, when combined with other non-manual features. Motivated by the domain of Sign Language Recognition, we formulate a classification problem for hand postures used in Greek Sign Language. More specifically, we aim to develop a lightweight, yet robust hand posture classifier. We present a method that processes unlabeled videos of subjects signing, and automatically assigns posture labels to the frames. This is based on using a 3D hand pose estimation tool, Google's MediaPipe (Zhang et al., 2020), for estimating the posture in each frame, and comparing the 3D configurations of joints to those corresponding to the problem's classes. Precisely because the annotation produced in this manner is contaminated with noise, this is a suitable problem on which to apply the techniques we present for handling noisy ground truth data.

2 RELATED WORK

This section briefly summarizes methods for the reduction of annotation effort, including machine learning approaches, and related ideas. A general technique which is comonly used both for annoatation effort reduction, and impeoved generalization is data augmentation (Shorten and Khoshgoftaar, 2019). A domain-specific technique is presented by Voigtlaender et al. (Voigtlaender et al., 2021), for the problem of semi-supervised video object segmentation. The authors propose a network to extract pixel-level pseudo-labels given bounding boxes.

Reducing annotation effort is also desirable in the field of active learning, the field of machine learning where an algorithm repeatedly queries a user, known as the oracle, for labeling new data. Sun et al. (Sun and Loparo, 2020) argue that even though several automatic or semi-automatic annotation methods can reduce the number of instances that need to be labeled, the queries to the oracle which offer the most to the learning algorithm remain the most difficult cases to label. The authors attempt to alleviate this issue by leveraging available metadata that can give the oracle "hints" for the labeling process, by clustering data points with similar metadata attributes. In contrast to active learning approaches, our work does not assume perfect on-demand annotation of samples, but rather estimates the most probable class label of samples, closer to the semi-supervised learning paradigm.

Semi-supervision refers to a family of machine learning techniques which operate on a small set of manually labeled data combined with a larger set of unlabeled data. These techniques are also relevant to the problem of reducing annotation effort. Semi-supervised learning refers to methods that are trained on a combination of a small amount of manually annotated data and a large amount of unlabeled data. Honari et al. (Honari et al., 2018) develop two techniques for landmark localization based on partially annotated datasets. The authors leverage the limited samples with landmark annotation, as well as a more abundant set of samples for which only a more general, high-level label is available. This label can be either for a classification or regression task, and serves as an auxiliary guide towards localization of landmarks on the unlabeled data. Wan et al. (Wan et al., 2017) present a semi-supervised approach for 3-D hand posture estimation from single depth images. The approach creates two generative models which share a feature space, such that any point in this space can be mapped to a unique depth image, and a unique 3-D hand pose.

Another approach related to semi-supervision and active learning is that of label propagation (Zhu and Ghahramani, 2002). Label propagation starts with a small set of labeled samples, and a larger, unlabeled set. The key idea is to progressively label samples from the unlabeled set, based initially only on the knowledge of the labeled samples, but gradually labeling more unlabeled ones, hence "propagating" the known labels. Our approach is similar to label propagation in the sense that it also begins by only using a small set of "trusted" samples. In contrast, however, we assume that all of the other samples are labeled as well, with labels that feature noise.

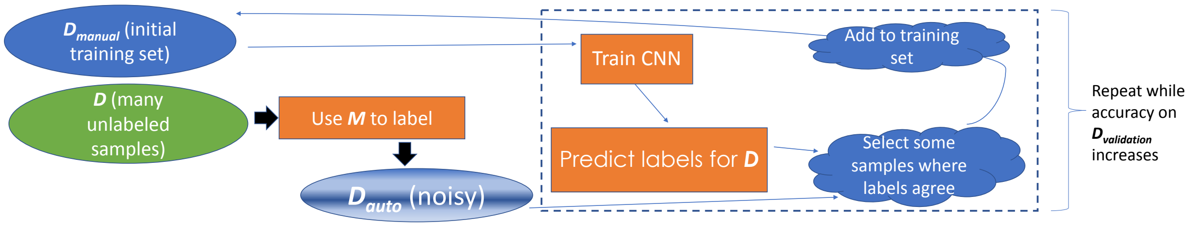
D_manual (initial training set)
D (many unlabeled samples)
Use M to label
D_auto (noisy)
Train CNN
Predict labels for D
Add to training set
Select some samples where labels agree
Repeat while accuracy on D_validation increases

Figure 2: Overview of the proposed approach. We assume the availability of a small, manually labeled and therefore reliable dataset D_{manual} (top left), and a larger unlabeled one D (bottom) left. Furthermore, an automatic method is assumed to provide noisy labels for D, yielding D_{auto} (bottom middle). We start by training a classifier on the reliable dataset D_{manual}, and proceed to iteratively expand the training set by adding samples from D where the prediction of the trained classifier agrees with the automatic annotation in D_{auto}.

An old paradigm in semi-supervised learning is self-learning, or self-training (Chapelle et al., 2009), which is based on the repeated use of a supervised method. Initially, the supervised algorithm is trained on labeled data only, and on each step some of the unlabeled data is labeled using the trained model. The procedure is repeated, and the training set gradually expands to feature data labeled by the algorithm itself. Also relevant to the topic of reduced annotation effort is the concept of learning from data where the ground truth is noisy. Automatically created ground truth has a greater chance of featuring noise. Therefore, techniques that tackle this issue also facilitate the use of automatic annotation methods. Jiang et al. (Jiang et al., 2018) develop a strategy for training on noisy ground truth based on Curriculum Learning. Curriculum Learning, developed by Bengio et al. (Bengio et al., 2009), is a technique for guiding the optimization of a neural network by presenting the training examples in an order which encourages it to progressively learn more complex features. Jiang et al. (Jiang et al., 2018) apply this concept by simultaneously training two networks, one which learns the actual task featuring the noisy ground truth, and another which learns how to guide the first by presenting samples that are deemed correct. While the second network undergoes a training process, the curriculum adapts to the data at hand. Hacohen et al. (Hacohen and Weinshall, 2019) investigate Curriculum Learning with two strategies. The first one involves a “teacher” network which transfers knowledge it has accumulated from some other dataset. The second is a type of bootstrapping, where the network is first trained on the target dataset without any curriculum. Han et al. (Han et al., 2018) also tackle the problem by coining the method “Co-teaching”. It involves training two networks simultaneously, where each one teaches the other about what data it considers correctly labeled. The intuition is that a network tends to learn correct samples in the first stages of optimization, and memorize the wrong ones at a later point (Arpit et al., 2017). Mirzasoleiman et al. (Mirzasoleiman et al., 2020) address the same issues by developing a method that can select subsets of the data that are likely to be free of noise. Their selection is based on inspecting the Jacobian matrix of the loss function being optimized, and choosing medoid data points in the gradient space. By choosing the samples in this fashion, they avoid overfitting on corrupted ground truth samples.

Li et al. propose DivideMix (Li et al., 2020), an approach to learn from noisy labels by leveraging semi-supervised techniques. Specifically, a mixture model on the loss is used to divide the training data into a labeled set with clean samples and an unlabeled set. Northcutt et al. propose Confident Learning (Northcutt et al., 2021), an approach to estimate the confidence/uncertainty of dataset labels. We choose this approach to experimentally compare against our work, since it is a recent, state-of-the-art work, with an easy to use and actively developed code base. Overall, on the topic of learning from noisy labels, Song et al. present a comprehensive overview (Song et al., 2020).

We propose an approach for working with noisy ground truth data that have been automatically annotated. Our approach exhibits similarities to the approaches presented here, especially to Curriculum Learning (CL) and self-training approaches. This is because it attempts to select appropriate subsets of the automatically annotated training data. Furthermore, this is done in an iterative manner, exploiting the predictions of the trained classifier itself. Overall, the most suitable learning paradigm fitting our approach is that of semi-supervised learning, and more specifically self-training. In contrast to self-training, our approach explicitly focuses on cases with large amounts of noisy data. To the best of our knowledge, no similar approaches on self-training with noisy labels or on related areas have been proposed in the relevant literature.

3 METHODOLOGY

In this Section we describe the proposed approach in detail, and also the method we followed to prepare data for the target task of hand posture recognition.

3.1 Exploiting Automatic Ground Truth

We assume a classification problem featuring K classes, and a method which is capable of automatically classifying a sample in one of these classes. However, it is assumed that this method is imperfect, yielding some erroneous annotations. It is expected that training a CNN model on a dataset with mislabeled samples will affect its ability to discriminate among the classes of the problem. It is therefore useful to investigate techniques that can take advantage of this noisy automatic labeling.

Among all candidate techniques, the simplest way to utilize automatically annotated data is to consolidate large numbers of such samples, potentially reducing the impact of failure cases on the performance of the classifier. One could also treat those samples as completely unlabeled, combining them with a small number of manually annotated samples and thus adopt a semi-supervised approach. However, it is our intuition that the utilization of a noisy ground truth label can be beneficial, and therefore a more sophisticated approach can be devised. Specifically we assume the availability of a large dataset, D_{auto} for which automatically produced ground truth labels are available. We also have a small set of samples, D_{manual} which we have annotated manually, and thus their labels are expected to be significantly less noisy. Training a model only on D_{manual} is likely to yield a classifier that does not generalize effectively. However, an important observation is that it can still provide a noisy estimate of the likelihood of a sample belonging in a particular class. For a novel sample, the commonly employed "one-hot encoding" for classification tasks will yield continuous values in the range $[0-1]$ (one per class), which can serve as estimates of such likelihoods.

Therefore, we can train a model on D_{manual}, compute its predictions on all samples of D_{auto}, and select only those samples where the prediction agrees with the labeling produced by automatic annotation. This yields $D_{auto,0} \subseteq D_{auto}$, a set of samples which are more likely to be correctly labeled, since we are conservatively selecting only the samples for which two noisy predictors agree. We expect a model trained on $D_{manual} \cup D_{auto,0}$ to generalize better than one trained on D_{manual} only, since it has a larger number of samples from which to extract useful classification features. Furthermore, the increased number of samples is more likely to prevent overfitting.

By continuing this iterative process, increasingly robust classifiers are formed, and we expect a better utilization of D_{auto}, since the combination of the two predictors will filter out some of the noise of the automatic annotation, as shown in Figure 2. Assuming a validation set, we can continue this iterative process until the validation accuracy of the classifier trained on the selected data no longer increases, or starts to decrease. The latter could potentially occur if the automatic labeling results in many similar samples with the same incorrect label. Even if only a few of those samples are added in the training set, the capacity of the model to memorize could result in more wrongly annotated samples "contaminating" the training set later on. We call this iterative scheme "Greedy Iterative Dataset Expansion Algorithm", or "G-IDEA".

Furthermore, we expect the classifier's predictions to become more trustworthy as the iterations progress, since it has more data available for training. Furthermore, as previously discussed, if many similar incorrect samples exist in D_{auto}, a model trained on a few of them can end up memorizing them. Subsequently, it is more likely to introduce more similarly incorrect samples in its dataset as the iterations progress.

Motivated by these observations, we can modify G-IDEA: On each iteration, only a portion of the data where the predictions agree with the automatic labeling is included in the training set, as illustrated in Figure 2. To select this portion, we assume that the model outputs K numbers representing the likelihoods of a sample belonging in each of the K classes. For each of the classes we can then order the predicted samples by decreasing likelihood. Intuitively, we are ordering the samples by a measure of how certain the model is of its predictions. We can then select only a conservative percentage of each class' data, and then gradually increase this percentage as the iterations progress. We call this modification "Conservative Iterative Dataset Expansion Algorithm", or "C-IDEA". Algorithm 1 outlines a simplified version of C-IDEA, with the addition of new samples performed over the whole dataset, for clarity of the presentation. This approach, C-IDEA, is the proposed method to exploit noisy, automatically annotated labels. Apart from motivating the development of C-IDEA, G-IDEA serves as a baseline in the quantitative evaluation.

3.2 Hand Posture Recognition

As already mentioned, the algorithms presented above are applicable in any classification problem.

Algorithm 1: Simplified version of C-IDEA: In practice the addition of samples is performed per class to avoid class imbalance in later iterations.

```
Input:
$D_{manual}$, a set of manually annotated samples
$D_{auto}$, a set of automatically annotated samples
$D_{validation}$, validation set, also manually annotated
$sel\_r$, initial training data selection ratio ( $\in (0,1]$ )
$inc\_factor$, ratio increase factor per iteration
Output:
$m$, a CNN model trained on selected input data
$D_{selected}$, the selected training data

  $m \leftarrow$ train a model on $D_{manual}$
  $D_{selected} \leftarrow D_{manual}$
  while accuracy on $D_{validation}$ increases do
    $new\_data \leftarrow$ all samples of $D_{auto} \setminus D_{selected}$ where the predicted class output of $m$ agrees with the label given in $D_{auto}$
    $l \leftarrow$ per sample likelihoods of all samples in $new\_data$, as estimated by $m$
    $new\_data \leftarrow$ sort $new\_data$ according to decreasing likelihood $l$
    $sel\_data \leftarrow$first $\lfloor sel\_r \cdot size(new\_data) \rfloor$ elements of $new\_data$
    $D_{selected} \leftarrow D_{selected} \cup sel\_data$
    $m \leftarrow$ train a model on $D_{selected}$
    $sel\_r \leftarrow \min(1, sel\_r \cdot inc\_factor)$
  $m \leftarrow m_{best}$ (the best performing model among all trained models according to validation accuracy)
  $D_{selected} \leftarrow D_{best}$ (similarly to above, the training set of $m_{best}$)
  return $m, D_{selected}$
```

For the current work, we choose to apply and evaluate them on the problem of hand posture recognition from RGB images. Given a single RGB image of a human hand, the task here is to output a label indicating which of the K postures appears on the image.

Experimental evaluation of the proposed techniques requires us to specify a set of hand postures which we are interested in recognizing. Motivated by the general problem of Sign Language Recognition, we apply our methods to a set of hand postures that convey semantic information in Greek Sign Language (GSL). Although recognition and translation of GSL cannot be performed with hand posture information only, the configuration of each hand can serve as a useful feature in the general task.

3.3 Automatic Hand Posture Extraction

As already outlined, the proposed approach assumes an automatic way to label a large part of the dataset, D_{auto}. Therefore, in the following we outline the approach we used to automatically label images of sign language. As already mentioned, this approach doesn't need to estimate perfect labels, in fact the annotation is assumed to be noisy. Our approach is based on extracting the 3-D keypoint structure of the hands. Then, each frame is assigned the label of the posture that best matches this structure, among a predefined set of postures we are interested in recognizing. In order to represent 3-D hand postures, we adopt a hand model commonly used in relevant literature (Panteleris et al., 2018; **?**; **?**). This model consists of 21 keypoints, the joints of the palm and fingers.

The first step in assigning one of K labels to any subset of the frames is extracting this 3-D structure for all classes and all frames. To achieve this, we used MediaPipe Hands (Zhang et al., 2020), a software developed by Google, capable of extracting 2.5-D hand landmarks from RGB images.

Given an input image of dimensions $W \times H$, the scheme (Zhang et al., 2020) utilized by Mediapipe represents each hand posture as a 21-tuple of 3-tuples,

$$P = ((x_1,y_1,z_1),(x_2,y_2,z_2),\ldots,(x_{21},y_{21},z_{21})).$$

All x_i, y_i are in the range $[0,1]$, and $x_i W$, $y_i H$ equal the horizontal and vertical pixel coordinates of landmark i in the image, respectively. Furthermore, z_i represents the relative depth of landmark i: the wrist joint is positioned at depth 0 by convention, and the remaining depths are appropriately assigned. In order to compare hand postures via their corresponding 3-D keypoints, we first have to transform all poses to a common coordinate system[1].

For translation and rotation normalization we assume that the wrist joint and the base joints of all fingers and thumb can be considered approximately rigid in the human hand. Let $P = (j_0, j_1, \ldots, j_{21})$ be a hand posture in an image of dimensions $W \times H$ as described previously, with $j_i = (x_i, y_i, z_i)$ corresponding to the i-th landmark as described above. Then, $P^{'} = (j_0 - j_0, j_1 - j_0, \ldots, j_{21} - j_0)$ is the same posture translated so that the wrist lies at the origin. We continue by rotating all the points of $P^{'}$ so that the vector from the wrist to the base of the index finger is lying on the x axis, and the palm is lying on a predefined plane. It is now partially meaningful to define the distance of two hand postures, $P_1 = (j_{1,1}, j_{1,2}, \ldots, j_{1,21}), P_2 = (j_{2,1}, j_{2,2}, \ldots, j_{2,21})$ as the sum of the Euclidean distances of all corresponding joints, as in Equation 1. However, the characteristics of each individual hand (e.g. finger lengths) can

[1] For this work, we make the simplifying assumption that no two classes can differ from one another solely by a common rotation of all their landmarks.

adversely affect this sum: two different hands performing the same posture may end up having a distance significantly larger than zero if their anatomical structures are different.

Therefore, it is necessary to also normalize the individual hand characteristics. To this end, we chose a predefined hand model, $H_0 = (j_{0,1}, j_{0,2}, \ldots, j_{0,21})$, and apply the following transformation on all postures, in order to match their structure with H_0. We first replace all the finger base joints in the normalized pose by their respective ones in H_0, effectively forcing an identical palm size. Then, for each "bone" (that is, segment of consecutive joints in the kinematic chain), we change its length to match the respective one of H_0, while preserving its direction in space.

We can now proceed to compare hand poses. To this end, as already mentioned, we employ the sum of euclidean distances between corresponding hand keypoints, after they have been normalized as above

$$d(P_1, P_2) = \sum_{k=1}^{21} \| j_{1,k} - j_{2,k} \|. \quad (1)$$

We now have a way of comparing every hand posture of every frame with the postures that correspond to the classes of our problem. More precisely, we are given K input images, each corresponding to a particular hand posture (that is, to a class of the problem) and a set of videos, each of them featuring one of subjects performing these hand postures. The end goal is to automatically label a subset[2] of the frames that can be then be used to train the classifier we are designing.

Towards this end, the K reference hand postures are compared with each of the frames to be automatically labeled, according to the metric of Equation 1. Each frame is then assigned the label of the reference pose with the lowest distance. The resulting dataset is termed D_{auto}. Furthermore, we proceed to manually or semi-automatically (with the aid of automatically extracted labels as above) annotate a small part of the available images into the datasets D_{manual}, $D_{validation}$, and D_{test}, paying attention to include images of different singers in each of them, excluding also the signers in D_{auto}.

4 EXPERIMENTAL EVALUATION

We present an evaluation of our proposed methods applied to the problem of hand posture recognition from RGB images. Firstly, we present the datasets that are used for model selection (validation set) and performance estimation (test set). We then present the details of the training procedure, with respect to the network's hyperparameters and data augmentation process. Finally, we compare the performance of CNNs trained with and without the use of the algorithms outlined in section 3.1.

4.1 Employed Datasets

In order to evaluate the proposed approach in real-world data we employ the HealthSign dataset, which is detailed in (Kosmopoulos et al., 2020). This is a dataset focused on communication of deaf patients with health professionals. From an analysis of the 450 most common glosses in the HealthSign dataset we found 38 postures being used at least once. From these 38 postures, we selected 19, for which the automatic annotation process had yielded some initial label, aiding the manual annotation process.

Apart from the images in the HealthSign dataset, more data were found from YouTube videos of signing subjects, as well as some more were captured by us. Seven male and three female volunteers performed the identified hand postures under our instructions, contributing to the available images. Overall, the HealthSign dataset features 8 signers, and from YouTube videos and our recordings we added another 7 and 10 signers respectively to the pool.

Among these, images from the HealthSign dataset were used for manual annotation, essentially exclusively populating the D_{manual} dataset. The remaining of the data were predominantly used for automatic labeling (populating D_{auto}) whereas some of the signers were held out, populating respectively either the $D_{validation}$ or the D_{test} parts of the dataset. Figure 3 shows three example images for posture "Y" from the test set, each from a different signer.

4.2 Classifier Architecture and Training

For all the experiments presented below, we choose the MobileNet v2 network (Sandler et al., 2019) architecture as the base for the classifiers we train. We choose it because it is a recent lightweight architecture that can be used in real-time conditions on smartphones, while also achieving high accuracy (e.g. on the ImageNet dataset (Deng et al., 2009)). In particular, we always start with a MobileNet v2 model that has already been trained on the ImageNet dataset. In order to adapt the model to our classification problem, we add a fully connected layer after the convolutional part of the MobileNet architecture. Each of its neu-

[2] In practice many of the frames of a video recording must be discarded because the signer is in the so-called "neutral pose". We resort to simple heuristics such as thresholds on hand motion to detect and reject such frames.

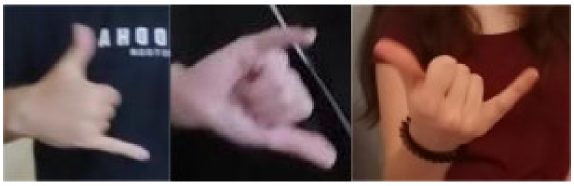

Figure 3: Three examples of images included in the test set for the posture "Y", from three different signers.

rons has as inputs all outputs of the final layer of the MobileNet architecture, and there are as many neurons as the K classes of the problem. Finally, the K outputs $(x_1, x_2, \ldots, x_K)$ are fed through a log-softmax layer since we are aiming for classification among K classes. During training, the objective function is the negative log likelihood loss function, which is commonly used for classification problems.

We tried both the AdaDelta (Zeiler, 2012) and Adam (Kingma and Ba, 2017) optimizers, with the PyTorch (Paszke et al., 2019) default learning rates 1 and 10^{-3} respectively. Preliminary experiments determined that the Adam optimizer generalized marginally better. The experiments detailed below were all performed with the Adam optimizer with this parameterization. During training we used a batch size of 64 samples. In every experiment presented below, the corresponding CNN was trained for 120 epochs. At the end of each epoch, the accuracy of the classifier on the validation set was measured, and the final model for that experiment was chosen to be the one which achieved maximum validation accuracy.

The fact that the classes are assumed rotationally invariant allows us to use a random rotation between 0° and 360° as augmentation. Since the images fed to the network are already tightly cropped around the hand, we avoid using random cropping as augmentation, since it could result in obscuring parts of the image that contain useful information for classification. We do, however, use a random translation augmentation of up to 10% of the image's dimensions. After applying all transforms, each image that is fed to the network is always scaled to be 224×224 pixels in size, and also normalized so that its pixel values have a mean of $\mu = (0.485, 0.456, 0.406)$ and standard deviation of $\sigma = (0.229, 0.224, 0.225)$ (values computed on the ImageNet dataset, and recommended for use with a pretrained MobileNet model).

For the case of training images that originate from the HealthSign dataset, we also apply one custom data augmentation step before applying any of the others described above. Specifically, because the HealthSign videos were recorded in a room with a mostly green background, we can determine an approximate pixel value (r_{bg}, g_{bg}, b_{bg}) for the background color in an offline preprocessing step. Then, during training, we can perform a crude segmentation of the background based on these color values. Having selected the background pixels, we then replace them with the corresponding pixels from an image randomly sampled from the Stanford backgrounds dataset (Gould et al., 2009). Figure 4 illustrates applying this transformation to an example image. We apply this transformation with a probability of 0.7 on any HealthSign training image that is fed to the network. This serves to provide greater variety to the HealthSign data, reducing the probability of overfitting on irrelevant features, such as learning to expect a green background around some or all hand postures.

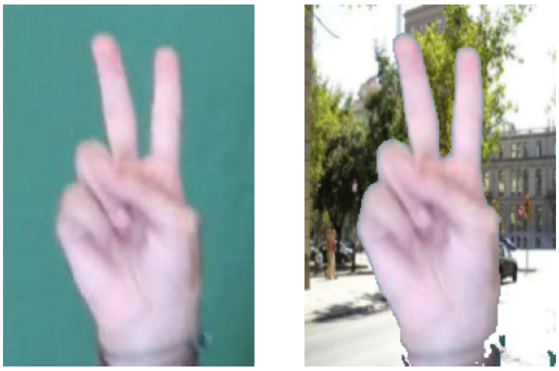

Figure 4: Transforming the background of HealthSign images. On the left, one of the original images. On the right, the same image augmented with a random background.

4.3 Experimental Evaluation

As a first, baseline experiment, we trained a classifier as outlined above, only on the manually annotated dataset D_{manual}. In this case, the trained model achieved an average accuracy of 46.5% on the validation set, and 41.5% on the test set.

Next, we experimented with the use of automatically annotated data as a complement for our small set of manually annotated images, naively adding them to the training pool. This is a large pool of samples from 17 signers. We progressively added more automatically annotated samples ordered by lowest distance (see Equation 1), only keeping the balance of samples per class and signer. We experimented with several values for the number of samples per class for every signer, beginning with 15 images per signer class, similarly to the manually annotated samples. By the term signer class we refer to all the images belonging in a particular class which feature the same signer. Specifically, we experimented with the values of 30, 60, 120 and 500 images per signer class.

Table 1: Validation accuracy achieved when using a variable number of images per signer class. Top row: Images per signer class, bottom row: Validation accuracy.

15	**30**	**60**	**120**	**500**
69.06%	71.33%	70.4%	72.3%	**73.38%**

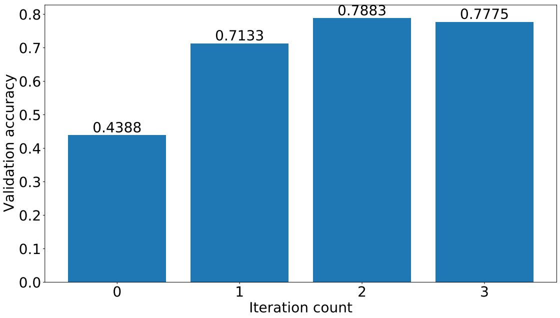

Figure 5: Validation accuracy achieved on each iteration when using G-IDEA.

Table 1 summarizes the highest validation accuracies recorded with each configuration.

We observe that, even with 15 images per signer class, there is a significant improvement (25%-30%) over using manually annotated data only. Furthermore, increasing the number of images per signer class is, in general, beneficial to validation accuracy. We record the highest validation accuracy of 73.38% when using 500 images per signer class. However, we also observe that the increase in accuracy is rather small in proportion to the increase in training set size. Considering that a large number of new, correctly labeled samples are introduced, one might expect a larger increase in accuracy. This hints that a significant amount of mislabeled samples have gradually been introduced in the dataset as well. We stopped experiments at 500 images per signer class because the increase in performance had almost plateaued at 500, and the additional computational cost didn't justify experimenting with larger values.

Next, we experimented with G-IDEA, serving as a baseline for C-IDEA. In this case, the training samples were iteratively selected by the proposed approach, as outlined in Section 3.1. For this reason, we anticipated the method to result in a less noisy dataset. Consequently, we also expected improvements on validation and test accuracy. Indeed, this was observed, as shown in Figure 5, depicting the evolution of validation accuracy on each iteration of the algorithm. Peak accuracy was reached on iteration 2, and the algorithm terminated on iteration 3, since the accuracy no longer increased. We ran the algorithm for two more iterations, and validation accuracy continued to decrease very slowly.

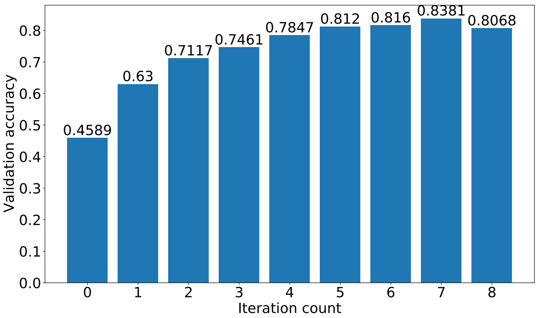

Figure 6: Validation accuracy achieved on each iteration when using C-IDEA.

Average accuracy was increased over the naive approach by approximately 5% both on the validation and test set. Furthermore, we observed in the respective confusion matrices that cases of confusion between specific classes decreased compared to the naive approach. Also, without the use of this algorithm the lowest class accuracy recorded on both validation and test sets was 25%, even if the average accuracy was in the order of 70%. When using G-IDEA, the lowest class accuracy was 44%. Filtering noisy ground truth samples therefore apparently contributes to the performance of a trained classifier, although there is still potential for further improvement.

Finally, the proposed approach, C-IDEA was experimentally assessed, which also selects new training samples iteratively, but makes its selections in a more conservative manner. We assessed whether this more conservative strategy could yield improvements. In particular, we used the aforementioned algorithm with parameters $sel\_rat = 0.1$ and $inc\_factor = 0.15$. The evolution of validation accuracy as the iterations progress is shown in Figure 6. Compared to G-IDEA, accuracy increased at a slower rate, which is to be expected, since the increase of training data is by design slower. However, from iteration 5 onward, C-IDEA achieved accuracies higher than the ones recorded in the previous experiment, culminating in a validation accuracy of 83.81%.

We recorded an accuracy of 83.81% on the validation set, and 85.72% on the test set, thus improving by approximately 5% over G-IDEA. The lowest class accuracy recorded on both validation and test set was now 65%, and we observed once again a decreased number of cases where one class was significantly confused with another. Therefore, C-IDEA appears capable of selecting mostly correct samples.

As a comparison with the state of the art, we performed an additional experiment using the Confident Learning method by Northcutt (Northcutt et al., 2021). In that experiment, the whole training dataset,

Table 2: Summary of validation and test accuracies achieved by each method.

Technique/data	Vali. accuracy	Test accuracy
Manual data	46.45%	41.52%
Manual & automatic data	73.38%	74.19%
G-IDEA (proposed)	78.83%	80.04%
C-IDEA (proposed)	**83.81%**	**85.73%**
Confident Learning (Northcutt et al., 2021)	73.58%	74.84%

$D_{manual} \cup D_{auto}$, was jointly provided to that method[3], which then decided which samples were to be trusted. The resulting selection was then used to train a classifier. The validation accuracy of this model was 73.58%, marginally better than the naive approach. Therefore, while it manages to slightly improve over the naive approach, evidently our proposed approach is better suited for the task at hand. One reason for this may be the fact that our approach treats differently the parts D_{manual} and D_{auto}, trusting the first to exploit the second.

The experimental analysis presented in this section proves that automatically annotated data can be highly beneficial in our problem. The mere inclusion of automatically labeled samples contributes significantly to the generalization of the network, increasing average accuracy on unseen data from the range of 40%-45% to 73%-74%. Furthermore, the cost in human effort for gathering the data is rather small. Additionally, the use of C-IDEA further increases accuracy to 83%-85%. These results are summarized in table 2, along with the performance of Confident Learning (Northcutt et al., 2021) for comparison.

5 DISCUSSION

5.1 Summary

We presented a method for utilizing automatically annotated data in training CNNs on classification problems. The method is based on training the network on a small subset of manually annotated data, and then iteratively adding samples that are likely correctly labeled. Iteratively, the network is retrained on the new training set, gradually becoming more accurate. We applied these techniques on the problem of hand posture recognition from RGB images.

5.2 Limitations

A limitation that stood out during the experimental evaluation of our approach was a sensitivity of the proposed approach in inherently ambiguous classes. Specifically for our test case, postures of different classes that nevertheless were similar, for example differing by the pose of a single finger, were more tricky to correctly classify. This may happen because an incorrectly labeled sample may lead to a cascading effect: after the network is trained with it, similar, incorrectly labeled samples enter the training set, perpetuating the initial classification error. Nevertheless, the proposed approach still outperformed the naive approach, probably because the selected correctly labeled samples outnumbered the incorrect ones in the later iterations. More generally, spurious entries in the early steps are problematic because they may lead to this cascading effect. A potential mitigating strategy would be to reevaluate the selection, and remove some of the least confident samples in each iteration.

5.3 Future Work

Among the numerous future directions of this work, a few stand out: Better heuristics to determine the most appropriate samples to add in each iteration can potentially yield further improvements. Also, the problem of hand posture recognition may benefit from different 3D hand pose estimation techniques, other than MediaPipe (Zhang et al., 2020), or even in conjunction with it. Another interesting direction to investigate is the possibility to train and use multiple classifiers, in a boosting fashion. This would allow for more reliable class predictions and possibly faster convergence of C-IDEA. Finally, problems other than hand posture classification are worth investigating.

ACKNOWLEDGEMENTS

This work is partially supported by the Greek Secretariat for Research and Technology, and the EU, Project HealthSign: Analysis of Sign Language on mobile devices with focus on health services T1EΔK-01299 within the framework of “Competitiveness, Entrepreneurship and Innovation” (EPAnEK) Operational Programme 2014-2020.

REFERENCES

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville,

[3] We used the implementation provided by the authors.

A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Gould, S., Fulton, R., and Koller, D. (2009). Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1–8.

Graves, A., rahman Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks.

Hacohen, G. and Weinshall, D. (2019). On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pages 2535–2544. PMLR.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Honari, S., Molchanov, P., Tyree, S., Vincent, P., Pal, C., and Kautz, J. (2018). Improving landmark localization with semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1546–1555.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Kosmopoulos, D., Oikonomidis, I., Constantinopoulos, C., Arvanitis, N., Antzakas, K., Bifis, A., Lydakis, G., Roussos, A., and Argyros, A. (2020). Towards a visual sign language dataset for home care services. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 520–524. IEEE.

Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Mirzasoleiman, B., Cao, K., and Leskovec, J. (2020). Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33.

Northcutt, C., Jiang, L., and Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Panteleris, P., Oikonomidis, I., and Argyros, A. A. (2018). Using a single rgb frame for real time 3d hand pose estimation in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV 2018), also available at CoRR, arXiv*, pages 436–445, lake Tahoe, NV, USA. IEEE.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2019). Mobilenetv2: Inverted residuals and linear bottlenecks.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48.

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2020). Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*.

Sun, Y. and Loparo, K. (2020). Context aware image annotation in active learning. *arXiv preprint arXiv:2002.02775*.

Voigtlaender, P., Luo, L., Yuan, C., Jiang, Y., and Leibe, B. (2021). Reducing the annotation effort for video object segmentation datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3060–3069.

Wan, C., Probst, T., Van Gool, L., and Yao, A. (2017). Crossing nets: Dual generative models with a shared latent space for hand pose estimation. In *Conference on Computer Vision and Pattern Recognition*, volume 7.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method.

Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.-L., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking.

Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.