Fine-grained Visual Categorization with 2D-Warping

Harald Hanselmann and Hermann Ney
Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
<surname>@cs.rwth-aachen.de

***Abstract*—The task of fine-grained visual categorization is related to both general object recognition and specialized tasks such as face recognition. Hence, we propose to combine two methods popular for general object recognition and face recognition to build a new model-free system for fine-grained visual categorization. Specifically, we use Local Naive-Bayes Nearest Neighbor as a pre-selection method and 2D-Warping as a refinement step. For the latter, we explore different ways to use the alignments computed by a 2D-Warping algorithm for classification. We demonstrate the performance of our approach on the CUB200-2011 database and show that our approach outperforms the recognition accuracy of current state-of-the-art methods.**

I. Introduction

While in general object recognition the goal is to find the general category of an image, e.g. does the image contain a car or a house or a bird, the sub-category is of interest in Fine-grained Visual Categorization (FGVC). For example if it is already known that the image contains a bird, the question is, what species is the bird? To answer this question is a challenging task, since the distinguishing features between classes can be very elusive.

One extreme case of FGVC is face recognition. Here, only very subtle cues define a class while for example changes in pose or facial expression cause large intra-class variability. However, this field is well studied [29] and recently convincing recognition results have been achieved by using 2D image-matching methods (*2D Warping*) with nearest-neighbor classification [18], [1], [11]. These algorithms compute an image matching for each pair of test and training images by minimizing an energy function including smoothness terms. This way, local 2D neighborhood dependencies between image descriptors can be accounted for. The resulting energy serves as a score measuring the quality of the matching and can then be used in a nearest-neighbor classifier. This approach is model-free and does not require any training. Since 2D-Warping (2DW) works well on face recognition, it is also a good candidate for solving more general FGVC tasks. However, a main drawback of 2DW-methods is that they are computationally expensive making them infeasible for large databases.

Another model-free algorithm based on local features is the *Naive-Bayes Nearest Neighbor* (NBNN) classifier [3], which was introduced in the context of general object recognition. In contrast to 2DW, this algorithm does not consider dependencies between image descriptors and works with image-to-class distances rather than image-to-image distances. By using KD-trees this can be implemented much more efficiently than the 2DW algorithms. In [16] the computational efficiency and recognition performance of NBNN is further improved by a variation of the algorithm which is based on local neighborhoods within a single search structure over the complete training set (LNBNN).

Considering the good recognition performance of 2DW and the efficiency of NBNN/LNBNN, we propose to combine these approaches to a model-free classification framework for FGVC. In this framework, LNBNN is used to find a pre-selection of classes which are likely to be correct. This is done in an attempt to significantly reduce the size of the training set used for 2DW in order to counteract the high complexity. Thus, applying 2DW in this manner can be seen as a refinement step. This is conceptually equivalent to pre-filtering in image retrieval (e.g. [19]), where a fast pre-filter is applied before using a more accurate ranking function. Additionally, we explore how the final matching alignments between test and training images computed by 2DW can be used for classification. Specifically, we aim to use image-to-class distances since this is known to perform better than image-to-image distances [3].

The paper is structured as follows. In Section II we give a short overview of the state-of-the-art in FGVC. This is followed by a brief review of NBNN/LNBNN and 2DW in Section III. A detailed description of our recognition framework is given in Section IV and an evaluation in Section V. Finally, we conclude our findings in Section VI.

II. Related work

Several different approaches to FGVC have been proposed in the past years. The work in [8], [27] tries to adapt the poselet-approach [4] known from body parts detection. In [2] part-based high-level features are learned. The work in [26] tries to avoid visual codebooks and human annotations by defining an image representation based on template matching. In [25] segmentation of semantic regions, hierarchical structure learning and geometric visual phrases are combined to one FGVC framework. The authors of [10] propose a rough alignment based on shape-masks obtained through segmentation. Some approaches are based on human feedback, e.g. the work in [22] where part-detectors and classifiers are refined by asking a user questions about part locations and attributes.

There is also a number of methods based on Deformable Part Models (DPM) [9]. In [5] the authors propose a model based on symbiotic segmentation and part localization using DPM, while the work in [17] detects a characteristic part of the object using DPMs and uses segmentation afterwards. Another example are Deformable Part Descriptors [28], where descriptors are computed after a pose-normalization step using DPMs. This approach was improved by using deep convolutional neural network features (DeCaf) [6].

However, none of these works explored, whether 2DW with full 2D descriptor dependencies is helpful for FGVC.

Noteworthy is also the work in [24] and [12], where image-to-class distances are proposed for face recognition. However, this is done with many restrictions and without considering 2D-dependencies. In [30] an image-to-class distance for image classification is introduced on a more abstract level. The authors represent an image class as a composition of objects and learn a distance based on matching these objects.

III. Background

For the rest of the paper, we assume we have given a set of training images $\mathcal{R}$, a set of test images $\mathcal{X}$ and a set of classes $\mathcal{C}$. Each class has N_c images. The goal is to find the classes of the test images. Extracting features with a dense grid on a test image $X \in \mathcal{X}$ leads to a set of D_X descriptors. The image is then represented as $X = \{x_{1,1}, ..., x_{I,J}\}$, where $x_{i,j}$ is the descriptor for grid position (i, j). Analogously a training image is represented as $R = \{r_{1,1}, ..., r_{U,V}\}$. Additionally, we have a function $d(x_{i,j}, r_{u,v})$ that measures the distance between two descriptors, e.g. the ℓ^1-norm or ℓ^2-norm.

A. NBNN/LNBNN

The NBNN algorithm proposed in [3] is a simple yet powerful classification algorithm.To classify a test image X, NBNN computes a score for each class $c \in \mathcal{C}$ by accumulating the distances of each descriptor d_i to it's nearest neighbor in class c. This defines an image-to-class distance since the match of each descriptor can be from anywhere within the class and not just a specific image. Each descriptor is matched independently implementing the Naive-Bayes assumption. This is summarized in the decision rule

$$\hat{c} = \underset{c}{\operatorname{argmin}} \left\{ \sum_{i,j} d\left(x_{i,j}, NN_c(x_{i,j})\right) \right\}$$

where $NN_c(x_{i,j})$ returns the nearest neighbor of $x_{i,j}$ for class c. By using KD-Trees (one for each class) the nearest neighbor search can be done relatively efficiently.

Recently, the work in [16] advanced this idea by changing the nearest neighbor search to a local procedure. Instead of finding the best matching descriptor of each class, the authors propose to search for the best k matches in the complete training set defining a local neighborhood of the descriptor. To get a score for class c the lowest distances to a descriptor of class c present in the set of best k matches are accumulated. In case there is no descriptor of class c in the set, the distance to the $k+1$-th descriptor is used. The advantage of this algorithm is that only one search structure is needed and the properties of KD-trees can be better exploited leading to a faster algorithm. Additionally, the authors observed an increase in recognition accuracy. For these two reasons we select LNBNN as pre-selection method in our framework.

B. 2D-Warping (2DW)

One drawback of the NBNN/LNBNN algorithm is the independence (Naive Bayes) assumption. If the descriptors of the test image are warped independently, it is difficult to capture the geometric structures in the image. One possibility to improve this is to use absolute warping penalties by appending the corresponding pixel coordinates to the descriptors [3], weighted by a factor α. This encourages descriptors to be warped to roughly the same position in the target image. This leads to smoother warping, but assumes that the images are well aligned. Another possibility is to add local context to the distance function as it is used by many of the 2DW algorithms [18], [11]. Instead of comparing two descriptors directly, the distances of two patches centered around the descriptor positions is computed.

In the case of face recognition, which can be seen as an extreme case of FGVC, 2DW algorithms have been successfully applied [18], [1], [11]. These algorithms aim to reach smooth deformations by explicitly considering the neighborhood dependencies in a 2D grid. In [18], [11] general 2DW is defined as follows. Given a test image X and a training image R the goal is to find a mapping $(i, j) \rightarrow (u, v) = w_{ij}$ which is minimal according to the energy function [18]

$$E(X, R, \{w_{ij}\}) = \sum_{i,j} \left[d(x_{i,j}, r_{w_{ij}}) + T(w_{i-1,j}, w_{ij}) + T(w_{i,j-1}, w_{ij}) \right] \quad (1)$$

where $T(\cdot)$ is a relative warping penalty or smoothness function. Hard geometric constraints can be integrated into the penalty that enforce monotonicity and continuity for the warping mapping [20]. These constraints are used in [18], [11], but lead to certain restrictions. For example, no mirroring and no rotations bigger than 90 degrees are possible. This means that either all variability must be covered in the training data or the data needs to be normalized.

In contrast to NBNN/LNBNN (which can also be written as energy function), the minimization of Formula 1 leads to direct image-to-image distances which can be used in the following 1-NN decision rule

$$\hat{c} = \underset{c}{\operatorname{argmin}} \left\{ \min_{n \in \{1,..,N_c\}} \left\{ \hat{E}(X, R_c^{(n)}, \{\hat{w}_{ij}\}) \right\} \right\} \quad (2)$$

where $\hat{E}(X, R, \{\hat{w}_{ij}\})$ is the minimized energy for matching X and R, $\hat{w}_{ij}$ is the corresponding warping mapping and N_c is the number of training images in class c.

Since general 2DW with full 2D-dependencies is NP-complete [14], some algorithms optimize this function by relaxing the problem (e.g. by discarding some dependencies), while other algorithms keep all dependencies and try to find an approximative solution [18], [11]. In particular, in [11] an algorithm based on dynamic programming in a two-level procedure (2LDP) was proposed. It works by sequentially computing hypotheses for the columns of the test image and selecting the best sequence of these hypotheses as solution. With this approach the authors achieved state-of-the-art results for pose-invariant face recognition tasks.

IV. Our Approach

Our approach combines LNBNN and 2DW. To classify an image we first apply LNBNN to find the m most likely classes. This is followed by a refinement step where we use

2DW on a training set reduced to the m most likely classes. In particular, we select the two-level dynamic programming algorithm proposed in [11] for 2DW, but also other algorithms could be used. The results obtained by previous work on FGVC (e.g. [8], [27], [2], [28]) suggest that a part-based approach is favorable. Also for LNBNN and 2DW, too much background information can significantly influence the performance negatively. For this reason, we adopt the part-based concept for 2DW and LNBNN.

A. Part-based Matching

We divide the object in P different semantic course parts (e.g. the head and the body of a bird [8]) described by rectangular bounding boxes and match each part independently. The resulting scores are then combined by a weighted summation similar to [30] (the parts in our case correspond to the objects in [30]). On the one hand, each part has a fixed weight τ_p to regulate the contribution of part $p \in P$. On the other hand the dimensions of the bounding boxes can be different for each test image and part which leads to non-comparable matching scores. Therefore the scores are normalized given the bounding-box dimensions. In case of 2DW this leads to new energies computed by

$$\hat{E}(X, R, \{\hat{w}_{ij}\}) = \sum_p \left[\frac{1}{D_{X_p}} \cdot \tau_p \cdot \hat{E}(X_p, R_p, \{\hat{w}_{ij}\}) \right] \quad (3)$$

which can be directly substituted into the inner minimization of Formula 2:

$$\min_{n \in \{1,..,N_c\}} \left\{ \sum_p \left[\frac{\tau_p}{D_{X_p}} \cdot \hat{E}(X_p, R_p^{(n)}, \{\hat{w}_{ij}\}) \right] \right\} \quad (4)$$

Note that Formula 4 ties the energies of matching all parts of the test image to the same training image. However, it is possible that better training matches for the different parts of the test image can be found in several different training images (e.g. the head of a bird in a test image is matched to one training image while the body is matched to another training image)[30]. For this reason the minimization over the training images is moved into the sum over the parts

$$\sum_p \left[\frac{\tau_p}{D_{X_p}} \cdot \min_{n \in \{1,..,N_c\}} \left\{ \hat{E}(X_p, R_p^{(n)}, \{\hat{w}_{ij}\}) \right\} \right] \quad (5)$$

This way, each part of the test image is warped independently and the best results are used for the final score. This is very similar to NBNN, but on part level (NBNN matches descriptors, while here parts are matched). The formula for LNBNN can be derived analogously.

B. Part-based best-k Accumulation (PartAcc)

The decision rule in Formula 5 describes the case of 1-NN. Naturally, this can also be extended for k-NN classification, where $k > 1$. However, one of the most important results of [3] is that image-to-class distances lead to a better performance than direct image-to-image distances. Thus, instead of computing the k nearest neighbors of the current part of the test image (X_p) and using a voting-scheme, we compute the best k matches for X_p in each class and sum up the corresponding scores to get one final image-to-class distance (c.f. Algorithm 1). Note that this is done after the 2DW alignments have been computed using 2LDP and for each part independently (within the summation over p in Formula 5).

Algorithm 1 Part-based best-k Accumulation (PartAcc)

Require: Test part X_p, training classes $\mathcal{C}$ with images $\mathcal{R}_p$, k and the warping algorithm $2LDP(X_p, R_p)$ that returns the energy cost of matching X_p to R_p

```
for C in 𝒞 do
    for R_p in 𝒩_p^(C) do
        image_scores[R_p] ← 2LDP(X_p, R_p)
    end for
    sort(image_scores, ascending)
    class_score[C] ← Σ_{R_p=0}^{k-1} image_scores[R_p]
end for
return class_score
```

C. Pixel-based best-k Accumulation (PixelAcc)

The part-based accumulation in Algorithm 1 uses the minimized energies for each part-matching (X_p to $R_p^{(n)}$) directly in the nearest neighbor classification. This means that for each energy all pixels from X_p must be matched to the same $R_p^{(n)}$. However, due to occlusions or pose-variations it might be better to split this up on the pixel-level. For this reason we propose a pixel-based best-k accumulation approach. For simplicity we again consider the case $k = 1$. When X_p is matched to a set of N_c training images of a specific class c, N_c different alignments are computed. Specifically, each pixel (i, j) in X_p has N_c different alignments $\hat{w}_{ij}^{(n)}$, one in each training image. For the case of PartAcc (c.f. Formula 5) when the costs for each pixel (i, j) are accumulated, the n in all $\hat{w}_{ij}^{(n)}$ must be the same. If we discard this restriction, each pixel in X_p can be matched to a different image. However, the warping penalty is only meaningful, if the neighboring pixels in X_p are matched to the same $R_p^{(n)}$. For this reason, we first transform Formula 5 by using only the descriptor distances and discarding the warping penalty from the accumulation (the previously computed 2LDP-alignments are still optimized including the penalty):

$$\sum_p \left[\frac{\tau_p}{D_{X_p}} \cdot \min_{n \in \{1,..,N_c\}} \left\{ \sum_{x_{ij} \in X_p} \left[d(x_{ij}, r_{\hat{w}_{ij}^{(n)}}) \right] \right\} \right] \quad (6)$$

Next, instead of combining all matches from one image to one final score and minimizing over those, we do the inner minimization of Formula 6 on pixel-level for each x_{ij}:

$$\sum_p \left[\frac{\tau_p}{D_{X_p}} \cdot \sum_{x_{ij} \in X_p} \left[\min_{n \in \{1,..,N_c\}} \left\{ d(x_{ij}, r_{\hat{w}_{ij}^{(n)}}) \right\} \right] \right] \quad (7)$$

Extending this to k-NN classification analogously to Section IV-B we get Algorithm 2. Note that this corresponds to using NBNN with a set of training descriptors that has been reduced by 2DW.

Algorithm 2 Pixel-based best-k Accumulation (PixelAcc)

Require: Test part X_p, training classes $\mathcal{C}$ with images $\mathcal{R}_p$, k and the warping algorithm $2LDP(X_p, R_p)$ that returns the energy cost of matching X_p to R_p for each pixel

for C in $\mathcal{C}$ **do**
  for R_p in $\mathcal{R}_p^{(C)}$ **do**
    $pixel_scores[i,j][R_p] \leftarrow 2LDP(X_p, R_p) \forall x_{ij} \in X_p$
  end for
  for $x_{ij} \in X_p$ **do**
    $sort(pixel_scores[i,j], ascending)$
  end for
  $class_score[C] \leftarrow \sum_{x_{ij} \in X_p} \sum_{R_p=0}^{k-1} pixel_scores[i,j][R_p]$
end for
return $class_score$

V. Experiments

We evaluate our approach on the CUB200-2011 database [23], which is quite popular in the FGVC community. The task is to classify images of birds according to their species. The complete database contains 200 classes and 11788 images. However, there is also a subset of this database used in the literature, where the data is reduced to all images of 14 classes (817 images total). We refer to this subset as CUB200-2011-14. We use the test-train split provided with the database which leads to roughly 30 images per class for training and another 30 images for testing.

For the part-based matching described in Section IV-A we use two coarse parts, the head and the body ($P = \{head, body\}$) as in [8]. We keep τ_{head} fixed at 1.0 and only alter τ_{body}. To apply the part-based matching bounding-boxes for the parts are needed. For this we use the part-annotations provided with the database. These annotations are more fine-grained with a total of 15 different parts. To get the coarse parts we combine the fine-grained parts beak, crown, forehead, left eye, right eye and throat to describe the head. For the body we use the parts back, belly, breast, left wing, right wing and nape. Additionally, we perform a normalization step based on the bounding boxes and six of the fine part annotations, which we will describe in detail in the next section. Since we are using most of the part-annotations, our setup corresponds to the localized species benchmark described in [23].

A. Normalization

Due to the geometric constraints of the 2DW algorithm, the normalization of the images is a crucial step in our setup. The goal is to roughly align the images, such that mirroring or large rotations are not necessary when matching them. At this point we need part annotations of all images (from the training *and* the test set). On the one hand, we use the annotations to get a bounding box of the head and the body. On the other hand, for each part we need three key locations to align the images by rotation. In case of the head these key locations are the left eye, right eye and the beak. The images are rotated such that either left eye and right eye, right eye and beak or left eye and beak are on a horizontal line (in that order). Additionally we make sure that the beak (if visible) is always on the left side by mirroring the images accordingly. In case less than two of the three key locations are annotated, the rotation step is skipped.

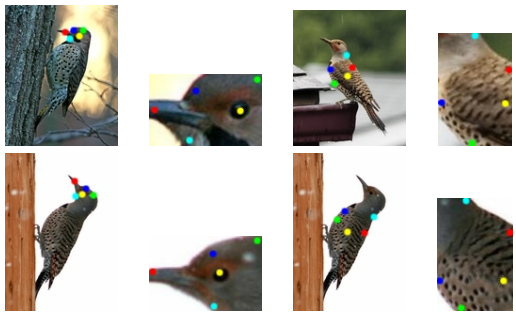

Fig. 1. Example of the normalization for the side view. The annotations of the parts are visualized by the colored dots. Left two columns: head-normalization; Right two columns: body-normalization.

In case only one part is annotated overall we select a bounding box of fixed size around this part. The same procedure is done for the body-part, except here the three key locations are left wing, right wing and breast. In the rare case that no part to construct a bounding box for a coarse part is annotated, this coarse part is discarded and the classification is based only on the remaining part.

Finally, the images are cropped according to the computed bounding boxes and resized, such that the bigger dimension equals 120 and the aspect-ratio is maintained. An example of the normalization procedure is given in Figure 1.

B. Features

Selecting good features is a very important step in a local descriptor based framework. In [2] a combination of color-histograms and histograms of oriented gradients is used with success, while the 2DW methods [18], [11] rely on PCA-reduced U-SIFT features [15], [13], [7]. In this work, we use a little bit of both. The color-histograms in [2] are computed by using k-means clustering to find 32 cluster centers as bins. The histogram is collected using an 8×8 or 16×16 neighborhood. For the first part of our feature vector, we do the same using the 16×16 neighborhood leading to a dimensionality of 32. For the second part of our descriptor we select the PCA-reduced U-SIFT features. However, color is a very important information for many FGVC tasks. Therefore, as in [8], we use a color-SIFT approach by extracting the U-SIFT features at each channel of the RGB color space as suggested in [21]. By reducing the vectors for each channel to a dimension of 30 using PCA, we get a final dimensionality of 90. Adding the color-histogram we obtain a final feature dimensionality of 122. These features are extracted based on a regular grid with step-size 3 and a padding border of 8 pixels at each side of the grid.

C. Results - CUB200-2011-14

First we evaluate our approach on the 14-class subset of CUB200-2011 using the software provided with [16] and [11]. All parameters are tuned empirically. As distance function for LNBNN and the 2DW algorithm 2LDP we use the ℓ^1-norm. The weight α for the absolute penalty for NBNN and LNBNN is 0.05 and the pixel coordinates are normalized. The nearest neighbor search is done with 4 randomized KD-trees (as it is done in [16]) and 4000 leaf-checks. The relative warping

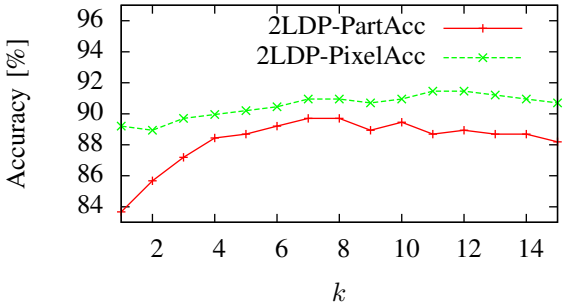

Fig. 2. Recognition accuracy for 2LDP with varying k and fixed τ_{body} (0.4 for 2LDP-PixelAcc and 0.9 for 2LDP-PartAcc).

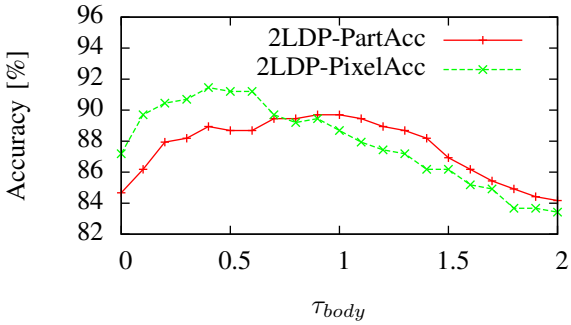

Fig. 3. Recognition accuracy for 2LDP with varying τ_{body} and fixed k (12 for 2LDP-PixelAcc and 8 for 2LDP-PartAcc).

penalty for 2LDP is the squared ℓ^2-norm. The local context for 2LDP are patches of size 7×7.

1) PartAcc vs PixelAcc: We start by comparing part-based and pixel-based accumulation. For this we run the full experiment without LNBNN-pre-selection and need to evaluate the parameters k and τ_{body}. By empirically tuning the two parameters the best result obtained for 2LDP-PartAcc is a recognition accuracy of 89.70%. The corresponding parameter values are $k = 8$ and $\tau_{body} = 0.9$. For 2LDP-PixelAcc the parameter set $k = 12$ and $\tau_{body} = 0.4$ gives the best results with a recognition accuracy of 91.46%. This means with the pixel-based accumulation we can improve the result by approximatively 2% absolute.

In Figure 3 the variation in accuracy for different values of τ_{body} are illustrated. For 2LDP-PixelAcc, the optimal value is approximately half of τ_{head} (which was fixed at 1.0). This means that the head-part contributes more to the recognition and the body is only an additional cue. This observation is supported by analyzing the special case of setting τ_{body} to 0 which corresponds to using only the head for recognition. With this setup 2LDP-PixelAcc already achieves very good results indicating that most of the recognition performance comes from the head. In fact, using only the body for recognition, we only get an accuracy of 62.56%. This is similar for 2LDP-PartAcc, where using only the body leads to a recognition accuracy of 59.05%. However, for 2LDP-PartAcc the graph reaches a peak at higher values with $\tau_{body} = \{0.9, 1.0\}$.

Figure 2 shows how the recognition accuracy varies if we keep the optimal value for τ_{body} fixed and alter the values of k. We can observe that 2LDP-PartAcc has a peak at $k \in \{7, 8\}$ but leads to stable results around 89% for $k > 4$. For 2LDP-PixelAcc the results are around 91% for $k > 4$ and reach a peak at $k \in \{11, 12\}$.

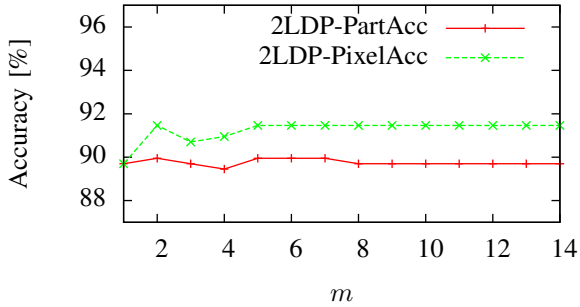

Fig. 4. Recognition accuracy for 2LDP with pre-selection by LNBNN.

2) LNBNN-pre-selection: Since we want to use LNBNN as a pre-selection method and 2LDP only for refinement, it is interesting to see how the recognition accuracy changes depending on the number of classes selected for refinement. This is illustrated in Figure 4 where we show the best recognition accuracies for 2LDP given the best m classes found by LNBNN. Starting with only one class for refinement (this corresponds to using just LNBNN) and adding more classes the recognition accuracy of 2LDP stabilizes for $m > 5$. This means by selecting the 5 best classes of LNBNN for refinement, no accuracy of 2LDP is lost, while the total runtime of 2LDP is reduced by a factor of $\frac{14}{5}$.

TABLE I. RESULTS ON CUB200-2011-14.

Method	Annotation	RA [%]
ℓ^1-norm	Part-Annotations	73.37
NBNN*	Part-Annotations	88.70
LNBNN*	Part-Annotations	89.70
2LDP-PartAcc	Part-Annotations	89.70
2LDP-PixelAcc	Part-Annotations	**91.46**
LNBNN + 2LDP-PartAcc	Part-Annotations	89.95
LNBNN + 2LDP-PixelAcc	Part-Annotations	91.46
POOF [2]	Part-Annotations	85.68
POOF [2]	Bounding-Box	70.10
Pose-Pooling [27]	Bounding-Box	57.44

* Numbers were obtained by running the code from [16] with part-based matching and our normalization and feature extraction.

3) State-of-the-art: In Table I we compare our approach to the state-of-the-art. For a fair comparison, it is important to note whether or not the fine part annotations (localized species benchmark) were used during testing. While we use the annotations for our normalization, some approaches use only the bounding boxes for the whole bird. Additionally to the results for LNBNN and 2LDP we also report results for NBNN and using the absolute distance in the same setup as 2LDP. Since these approaches also lead to good results we can conclude that the features and the normalization work very well on this dataset. For other state-of-the-art methods, the authors of [2] report results for both fine part-annotations and bounding-box annotations. To the best of our knowledge, their approach achieved the best overall recognition accuracy so far with 85.68%. We are able to improve this error rate significantly to 91.46% by using 2LDP-PixelAcc.

D. Results - CUB200-2011

The results for CUB200-2011 with all 200 classes are shown in Table II. While for CUB200-2011-14 applying 2LDP

without pre-selection is still feasible, on the complete dataset this is not the case (in our setup one matching takes roughly one second). For this reason we use 2LDP only as refinement on the m best classes found by LNBNN (the best result was obtained for $m = 4$). Again, the parameters τ_{body} and k are tuned empirically, but the other parameters are the same as in Section V-C. For LNBNN and 2LDP-PixelAcc the body-part has an increased importance and $\tau_{body} = 0.7$ is used for LNBNN and 2LDP-PartAcc while $\tau_{body} = 0.6$ is used for 2LDP-PixelAcc. Also for k there are differences compared to CUB200-2011-14. For 2LDP-PartAcc $k = 4$ leads to the best result and for 2LDP-PixelAcc it is $k = 2$. The much lower values of k indicate that there is much more intra-class variation in the full CUB200-2011 dataset. However, similar to CUB-2011-14, 2LDP-PartAcc achieves a similar result as LNBNN and 2LDP-PixelAcc outperforms both by approximately 2% absolute. Also on this dataset our approach leads to the best results on the localized species benchmark, although the margin to [2] is very small.

TABLE II. RESULTS ON CUB200-2011.

Method	Annotation	RA [%]
NBNN*	Part-Annotations	71.14
LNBNN*	Part-Annotations	71.25
+ 2LDP-PartAcc	Part-Annotations	71.64
+ 2LDP-PixelAcc	Part-Annotations	**73.71**
POOF [2]	Part-Annotations	73.30
Symbiotic [5]	Part-Annotations**	69.5
HPM [25]	Part-Annotations	66.35
DPD-DeCaf [28], [6]	Bounding-Box	64.5
Unsupervised alignments [10]	Bounding-Box	62.7
Symbiotic [5]	Bounding-Box	59.4
POOF [2]	Bounding-Box	56.78
Pose-Pooling [27]	Bounding-Box	28.18

* Numbers were obtained by running the code from [16] with part-based matching and our normalization and feature extraction.
** Only for the head.

VI. CONCLUSION

In this paper we proposed to combine LNBNN and 2DW (specifically 2LDP) to one system for FGVC. While LNBNN is used as a pre-selection method, 2LDP is used in a refinement step. For the latter, we have evaluated how the 2LDP-alignments can be used for classification. Based on a combination of color-histograms and PCA-reduced U-SIFT features we demonstrated the effectiveness of our approach on the CUB200-2011 database using the localized species benchmark. The features combined with the normalization already lead to good results for simpler methods underlining the usefulness of part-locations. However, for the localized species benchmark the proposed 2LDP-PixelAcc leads to the best recognition performance.

A possibility for future work is given by the part-annotations. For our approach we need the bounding box and three key fine parts for each coarse part for normalization and a rough alignment. At this point, a part-detector could be applied leading to a fully automatic system. Since the 2LDP offers flexibility to some extent, this appears to be a promising approach.

ACKNOWLEDGMENT

The authors would like to thank Jens Forster from the HLTPR Group at RWTH Aachen University for his useful comments and suggestions.

REFERENCES

[1] S. R. Arashloo, J. Kittler, and W. J. Christmas. Pose-invariant face recognition by matching on multi-resolution mrfs linked by supercoupling transform. *CVIU*, 115(7):1073–1083, 2011.
[2] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, pages 955–962, 2013.
[3] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, pages 1–8, 2008.
[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, pages 1365–1372, 2009.
[5] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328, 2013.
[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.
[7] P. Dreuw, P. Steingrube, H. Hanselmann, and H. Ney. Surf-face: Face recognition under viewpoint consistency constraints. In *BMVC*, pages 1–11, 2009.
[8] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, pages 161–168, 2011.
[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8, 2008.
[10] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, pages 1713–1720, 2013.
[11] H. Hanselmann, H. Ney, and P. Dreuw. Pose-invariant face recognition with a two-level dynamic programming algorithm. In *Pattern Recognition and Image Analysis*, pages 11–20. Springer, 2013.
[12] Y. Huang, D. Xu, and T.-J. Cham. Face and human gait recognition using image-to-class distance. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(3):431–438, 2010.
[13] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR*, volume 2, pages II–506, 2004.
[14] D. Keysers and W. Unger. Elastic image matching is np-complete. *Pattern Recognition Letters*, 24(1):445–453, 2003.
[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
[16] S. McCann and D. G. Lowe. Local naive bayes nearest neighbor for image classification. In *CVPR*, pages 3650–3656, 2012.
[17] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012.
[18] L. Pishchulin, T. Gass, P. Dreuw, and H. Ney. Image warping for face recognition: From local optimality towards global optimization. *Pattern Recognition*, 45(9):3131–3140, 2012.
[19] L. Torresani, M. Szummer, and A. Fitzgibbon. Learning query-dependent prefilters for scalable image retrieval. In *CVPR*, pages 2615–2622, 2009.
[20] S. Uchida and H. Sakoe. A monotonic and continuous two-dimensional warping based on dynamic programming. In *ICPR*, pages 521–524, 1998.
[21] K. E. Van De Sande, T. Gevers, and C. G. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.
[22] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, pages 2524–2531, 2011.
[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
[24] X. Wei, C.-T. Li, and Y. Hu. Face recognition with occlusion using dynamic image-to-class warping (dicw). In *FG*, pages 1–6, 2013.
[25] L. Xie, Q. Tian, S. Yan, and B. Zhang. Hierarchical part matching for fine-grained visual categorization. *ICCV*, pages 1641–1648, 2013.
[26] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, pages 3466–3473, 2012.
[27] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, pages 3665–3672, 2012.
[28] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, pages 729–736, 2013.
[29] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.
[30] G.-T. Zhou, T. Lan, W. Yang, and G. Mori. Learning class-to-image distance with object matchings. In *CVPR*, pages 795–802, 2013.