BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

COMPUTER VISION WITH ACTIVE LEARNING

POČÍTAČOVÉ VIDĚNÍ S AKTIVNÍM UČENÍM

PHD THESIS

DISERTAČNÍ PRÁCE

AUTHOR **MARTIN KOLÁŘ, Ph.D.**

AUTOR PRÁCE

SUPERVISOR **prof. Dr. Ing. PAVEL ZEMČÍK**

ŠKOLITEL

BRNO 2020

Abstract

Machine Vision methods benefit from improving models, tuning trained parameters, or labeling representative data. In a series of experiments, this work validates the hypothesis that Active Learning improves the accuracy of these models. By extending the pseudolabel framework to Active Learning, this work includes a One-shot-learning approach to learn novel image categories by utilising an algorithmic recommender, an online Graphical User Interface to optimise the online Exploration/Exploitation tradeoff for tagging, and a two-step offline binary Active Learning framework to improve the quality of data used for Font Capture. By demonstrating the benefit of Active Learning in these approaches, this work contributes to the hypothesis, as well as concrete Machine Vision applications.

Abstrakt

Metody strojového vidění se zdokonalují zlepšením modelů, laděním trénovaných parametrů nebo anotací reprezentativních dat. Tato práce řadou experimentů potvrzuje hypotézu že aktivní učení zvyšuje přesnost těchto modelů. Rozšířením přistupu pseudolabelů o aktivní učení přispívá tato práce přístupem „one-shot-learning“ k učení nových kategorií obrazů s použitím algoritmických doporučení, dále online grafickým uživatelským rozhraním pro optimalizaci dilema Exploration/Exploitation pro online tagování, a dvoukrokovým offline binárním přístupem aktivního učení pro zlepšení kvality dat používaných pro snímání fontů. Tím, že demonstruje přínos aktivního učení v těchto přístupech, přispívá tato práce k hypotéze i konkrétním aplikacím strojového vidění.

Keywords

Computer Vision, Object Classification, Semi-supervised Learning, Active Learning, Transfer Learning

Klíčová slova

Počítačové vidění, Detekce objektů, Částečné učení s učitelem, aktivní učení, přenášené učení

Reference

KOLÁŘ, Martin. *Computer Vision with Active Learning*. Brno, 2020. PhD thesis. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Dr. Ing. Pavel Zemčík

Computer Vision with Active Learning

Declaration

I declare that this dissertation thesis is my original work and that I have written it under the supervision of prof. Dr. Ing. Pavel Zemčík. All sources and literature that I have used during elaboration of the thesis are correctly cited with complete reference to the corresponding sources.

. . . .

Martin Kolář
September 14, 2020

Acknowledgements

I would like to thank my friends, colleagues, family, and supervisor for their unending support.

Contents

CHAPTER 1

Introduction

Machine Learning in general and Computer Vision in particular are highly sensitive to the amount and quality of data presented. Reproducible improvements in results and abilities of created models can be wholly attributed to the following factors: a model's prior and ability to fit, a model's actual fit to data, and representativity of training data.

With the vast majority of research focused on the first factor, and the vast majority of engineering work focusing on the second, the third factor receives far less attention. As quality data for various critical tasks is known to be costly to create, this work focuses on how this can be done most effectively. The optimisation of expert labelling and assessment of the resulting benefits is achieved via Active Learning.

A hypothesis regarding the benefit of Active Learning in Machine Vision is experimentally validated under these scenarios: when the labelling oracle is a human expert, and when the oracle is a pre-trained algorithm for another task. A large dataset, such as personal photos or hundreds of thousands of fonts, is most effectively labelled by using a human-in-the-loop approach. Facilitated by per-sample certainty analysis during training, the expert labelling effort achieves higher label quality with significantly fewer annotations. The second case, where existing models can preprocess useful information, requires systematic labelling and confidence tracking, or practical mappings. I have successfully used these approaches in my own published work, as presented in this thesis.

The hypothesis is experimentally validated in several contemporary scenarios, such as fully automated labelling (section 3.2) to create a dataset with combined information not available elsewhere, but with models trained on several other datasets. My work demonstrates that filtering data by retraining a model to select valuable examples only is shown to be usable to minimise human effort in creating new, useful datasets, and models.

In addition to four cases validating the hypothesis that Active Learning benefits contemporary Computer Vision, at the core of this work lies another practical contribution: an improvement to the accuracy to work ratio achieved through a new pseudolabels Active Learning framework to integrate labelling by existing models, human experts, and a trained agent.

CHAPTER 2

Hypothesis & Contribution

Generative and discriminative algorithms in Computer Vision are designed and trained to maximize their ability to generalize. This is tested on unseen data, and maximized by improving the model prior, improving the quality of the labelled and unlabeled data used, and by hyperparameter tuning. A tangential approach, compatible with improving the model and hyperparameters, is Active Learning, by which the gathered data is improved during training-time by effectively managing the labelling and selection effort of an expert.

2.1 Hypothesis

As it is desirable to optimally transfer knowledge and existing models to new tasks where data is limited, the following hypothesis is put forward:

> *Human and algorithmic expert annotation using Active Learning improves the accuracy of contemporary Computer Vision methods.*

The hypothesis is put forward with the expected result of achieving a significant margin, thus being both useful and demonstrably achievable. The term *contemporary* is taken to mean algorithms performing state-of-the-art accuracy in the case of discriminative models, or output quality for generative models, but not expecting to produce comparable results to next-generation models. The hypothesis refers to *human and algorithmic experts* because Active Learning is expected to be applicable both in the context of green-field datasets and applications and in Transfer Learning applications where one or more useful weak learners already exist. The concepts of *accuracy*, *quality*, and *labor time* are context-specific, referring to a context-dependent relevant applicable metric of generalization, and to the time and effort taken to achieve said results.

2.2 Method

The hypothesis is tested by comparing the achieved accuracy of this approach, as opposed to hyperparameter tuning, model tuning, and extended training time. These alternative approaches are known to hold the potential to improve accuracy and reduce labour time for Machine Vision tasks, and they are widely studied elsewhere [15, 35, 32, 14, 5, 27, 26, 20, 17]. The experimental proofs presented further are divided by approach and refer to my own relevant published work. These are experiments to demonstrate the benefit of Active Learning via sample selection, via human and algorithmic labelling, via feature selection and adaptive visualization, and pseudolabels.

The standard supervised, unsupervised, or semi-supervised setting can be formulated as training a model M with parameters ρ on two sets of data:

$$\rho = \underset{\rho}{\operatorname{argmin}} M(X_K \cup X_U) \tag{2.1}$$

Where X_K are datapoints with known labels, and X_U datapoints with unknown labels. The Active Learning approach has the additional $X_{C,i}$ subset of X_U, which contains the datapoints chosen to be labeled:

$$\begin{gathered} X_{C,i} = S_i(M, \rho_i, X_{K,i}, X_{U,i}) \\ \rho_i = \underset{\rho}{\operatorname{argmin}} M(X_{K,i} \cup X_{U,i} \cup X_{C,i}) \end{gathered} \tag{2.2}$$

And the hypothesis is therefore simply that there exists a selection mechanism S such that

$$\forall i \exists S \quad | \quad L(M(X_K \cup X_U)) < L(M(X_{K,i} \cup X_{U,i} \cup X_{C,i})) \tag{2.3}$$

Where L is an appropriately chosen loss function for the given model.

The results of the demonstration of the hypothesis have a wide array of applications, such as Association-rule Learning used by comparing the quality of texture synthesis algorithms over inputs with selected properties [KDC17, KCD15].

In the following chapter, I will test the hypothesis in various scenarios. These tests correspond to various perspectives under which Active Learning can be utilised, namely online training with an optimized graphical interface [HKL+12] in section 3.3, iterative dataset optimisation [KHZ20] in section 3.1, and active transfer learning with an algorithmic expert [KHZ16] in section 3.4.

2.3 Contribution and Proof Outline

The theoretical contribution is that these results irrevocably demonstrate that the systematic application of Active Learning improves the accuracy of contemporary Computer Vision models. A practical contribution is also made, in the form of an algorithmic process usable for Zero-shot Learning and One-shot Learning for image classification, given that a weak retrieval system is available from a large set, such as online image search indexing the Web.

Furthermore, applied contributions are made in the fields where the hypothesis has been tested: an improved dataset and algorithm for Font Capture and a GUI for image tagging with Active Learning.

Specifically, the existence claim of equation 2.3 is demonstrated by finding S for various scenarios of (L, M, X). For each, the contemporary Machine Vision approach is considered in comparison with the Active Learning alternative. These two options are then compared, with the desired objective of showing

$$\exists S \quad | \quad (L, M, X) \tag{2.4}$$

under the conditions of equation 2.3. By demonstrating this for four important problems of current research, the hypothesis is validated in a limited context. In order to validate the hypothesis with respect to all problems of Computer Vision, the argument by analogy is made that every Computer Vision problem can be aided by applying these Active Learning principles.

The cases under consideration are:

1. Tagging of closed-domain information by optimizing an Active Learning Graphical User Interface

2. One-shot learning with pseudolabels and a weak algorithmic expert

3. Active Learning for human experts to create data for Generative Adversarial Neural networks

4. Transfer Learning of algorithmic experts for Generative Adversarial Neural networks

These four cases correspond to sections of the following chapter, where they are treated in more detail. An overview of the interconnections of these sections and how they jointly support the thesis is as follows: Active Learning benefits the model and users directly by allowing them to more efficiently label data with general classes, as well as domain-specific information (1. - published in 2012). Moreover, sufficiently well-preprocessed image data allows high-quality training of classes without any human expert, by using an online image search algorithm and pseudolabels to train a Convolutional Neural Network (2. - published 2016). The filtering of existing datasets by humans (3. - under review), as well as the labelling of unsupervised datasets by algorithms (4.) can be performed with Active Learning, enabling improved generative quality as well as entirely new applications.

Therefore, these four problems present a holistic approach to the application of Active Learning in contemporary Computer Vision. Beyond the contribution made to the hypothesis, these have also served to further the fields of research they have been applied in, as detailed in the following chapter.

CHAPTER 3

Active Learning for Machine Vision

The cases in which Active Learning has been tested to support the hypothesis are described in this chapter. The presented work is divided into four sections, loosely corresponding to my own published work. By using a combination of Uncertainty Sampling and Diversity Sampling, the sections below focus on creating improved data and models by sampling from all datapoints while optimizing labelling. Each of the following four cases is an experiment to test the hypothesis, and thus to serve as a quantitative proof.

In the context of labelled image datasets, image-wise tagging is not limited to pre-training annotation. In fact, the required expert input can be reduced by judicious initialization with an external system [KHZ16], by asking the annotator to verify rather than label [15], and by in-the-loop training to identify samples with low certainty [HKL+12].

This chapter describes my own work, in which the first approach has been tested and published, as detailed in section 3.4, the second approach has been tested in the context of generating fonts 3.1, and the last approach has been experimentally validated through implementation and user experiments 3.3. Similarly to the first approach, work in section 3.2 also shows that beneficial results can be achieved with pure Transfer Learning, where a set of labels is created from specialized pre-trained models, serving new tasks not possible before. Section 3.2 presents work made public as a freely available dataset at https://github.com/DCGM/ffhq-features-dataset.

These three of sections correspond to peer-reviewed work, as follows: Section 3.1 contains work currently under review at The Visual Computer as *Font Capture in the Wild* [KHZ20], section 3.3 was published as *Annotating images with suggestions—user study of a tagging system* [HKL+12], and section 3.4 describes the method published as *Deep learning on small datasets using online image search* [KHZ16].

Finally, section 3.5 summarizes how these individual contributions support the hypothesis of the thesis, and integrates the findings into a cohesive methodical validation.

3.1 Improvements via Dataset Size and Quality

Font Capture is a task in Computer Vision and Computer Graphics, in which text present in an image is replaced with new text in the same font. Worldwide, 750 million people are native speakers of a language written in a Latin-derived alphabet with diacritics such as accents, subscripts, and superscripts [4]. However, out of an estimated 100 thousand digital fonts widely available, only a few hundred include these non-English characters.

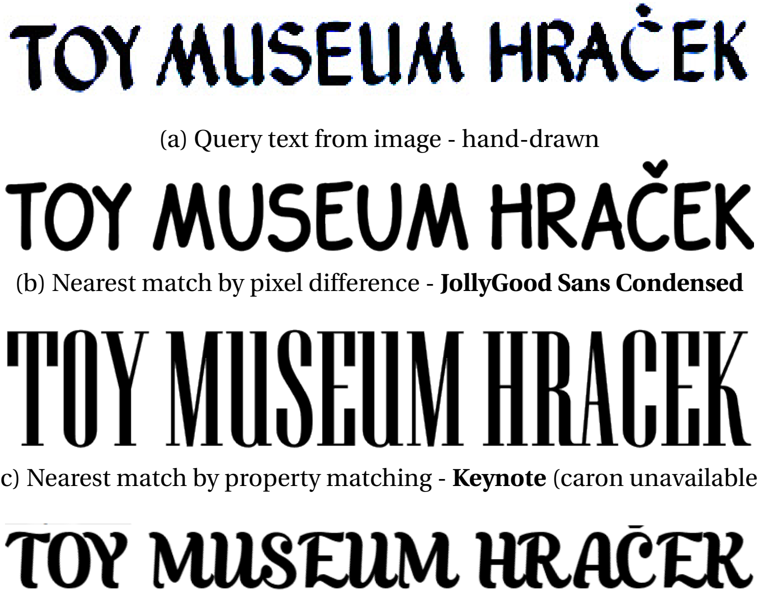
Font extraction on characters of the Latin alphabet has been attempted before, either with limited applications to classical fonts [11], or with blurry or noisy results [9, 33], and always by using individual characters as input. Thanks to an improved dataset and method for generating training samples, this work creates sharp fonts extracted directly from a line of text, suitable for use in photo editing as well as vectorization. This approach makes it possible to take an existing TrueType font, render new characters, convert them to vector graphics, and incorporate them in the original, thus effectively closing the loop.

Active Learning has been used to create a large high accuracy dataset of fonts, thus improving the quality of the method.

Generating fonts cannot be replaced by font search over a large enough dataset, as shown. Although fonts are widely shared on the internet, and font search engines are freely available, few fonts can be acquired to perfectly match the desired input. Finding a font given an image is a challenging task, undertaken by domain experts or automated processes. Identification methods range from pixel differences on detected aligned characters [2] to matching manually entered detailed features [3] based on standard font classification techniques [13], or automatically extracted attributes [23]. If these methods fail, fonts can be identified by a community of experts, such as Fontid.co. However, exotic fonts may be unknown to experts, unavailable to identification systems, or non-digitized. For example, Figure 3.1 shows a query text, along with nearest retrieved fonts by existing methods. This demonstrates that pixel difference and others are not a sufficient metric in font style matching.

These limits of finding existing fonts sparked an interest in extrapolating the entire style of a font from a single example. Font extrapolation with warp mappings dates to the nineties [30], inspired by the effect on the shape of charge on ink particles. A manifold over fonts has allowed smooth traversal of the font space [11] and was applied to classical typefaces to interpolate fonts. Extrapolation of numerals on the MNIST and SVHN datasets was made possible by deep generative models, creating a latent space which allows traversal across glyphs [18].

More recently, a fully connected deep net has been used to create an embedding of 50 thousand fonts [10]. A feed-forward neural network has been used to generate the entire font from four characters [9], with poor quality results. In addition to limited quality, this technique suffers from requiring specific characters, which may not occur in the sample text. Variational Autoencoders have been used to generate fonts from a single example glyph [33], but with a small dataset of 1’839 fonts and a fully connected network, the results are



(a) Query text from image - hand-drawn

(b) Nearest match by pixel difference - **JollyGood Sans Condensed**

(c) Nearest match by property matching - **Keynote** (caron unavailable)

(d) Nearest match by expert community - **Krinkes**

Figure 3.1: Comparison of font retrieval methods

still blurry. The 50k fonts dataset [10] has been used to train a VAE and a GAN [1], using the principles outlined in [26]. Fonts are extrapolated from varying characters with a Multi-Content GAN [7], in colour. However, existing methods require segmented characters rather than analyzing text directly. Most crucially, results of all existing methods are blurry or noisy for all but the most standard fonts.

While the existing methods train various architectures of neural networks with millions of parameters, I anticipate that increased quality may be reached through the application of Active Learning to create a larger, more representative dataset on which similar methods may be trained. The dataset was made by assembling a large pool of .ttf font files, iteratively training and annotating data for a binary classifier of usability, and thresholding the ensuing classifier to produce a dataset of usable quality fonts. This procedure utilizes a combination of Uncertainty Sampling and Diversity sampling, by focusing the annotator's attention on cases with high certainty, as well as cases of very low certainty.

The dataset is filtered through a shallow Convolutional Neural Network over three iterations. At each iteration, four representative characters of every font in the unlabeled dataset are rendered, classified, some are annotated, and the process repeats. The representative characters are „a“, „l“, „1“, and „?“.

The initialization proceeds as follows: The representative characters are rendered for all fonts and placed into a single image named with the unique font ID. If any of the characters „a-z“, „A-Z“, and „0-9“ is blank or undefined (rendered as in figure 3.2), the font is discarded immediately. Similarly, if any two characters are equal pixel-for-pixel, the font is discarded. All remaining fonts are viewed in a directory, allowing quick preview and easy group selection.

Figure 3.2: Undefined characters render as Unicode error codes

The fonts which do not contain readable Latin characters are manually selected and labelled as negative. This is performed for 0.5% of the data, or 1 300 fonts, which requires about two hours of annotation time. The other seen examples are marked as positive. This annotated data is then used to train a shallow Convolutional Neural Network. The network, used to classify usable fonts on four characters of fixed size, has two convolutional layers of 8 and 2 channels, and a last dense layer with a sigmoid activation function. This simple network is trained on the annotated data. Negatively annotated fonts include non-Latin fonts, dingbats, emojis, and highly ornamental typefaces, which may produce unexpected characters for standard glyphs.

Then, the network is used to make predictions on the unlabeled data. The 99.5% of unlabelled data receives ratings from 0 to 1, for which two tasks are semi-manually performed: the establishment of a threshold τ_1 where positive examples outweigh the negative, and manual labelling of unlabeled fonts near this threshold (uncertainty sampling), and near the 0 and 1 ratings (diversity sampling).

This for of uncertainty sampling is very effective, producing a high percentage of samples to be re-labelled. On the other hand, this simplified form of diversity sampling does not produce many examples to be re-labelled. This can be interpreted in two ways:

1. The classifier is very effective and has few high-certainty incorrect cases

2. This diversity sampling method is not effective at finding new types of cases needing re-labeling

After three iterations, the re-labelled fonts are once again thresholded with τ_3, and the effectivity of the combination of the sampling methods is evaluated as follows. A random sample of positive fonts is taken and manually evaluated until a false-positive is present (an incorrect font selected as correct). Using this method, the first false positive in random data was found at position 349, giving an expected accuracy of over 99.3%.

The dataset is then processed further, to create a specialized section of fonts with diacritical marks. This process is performed as in the initialization stage of the full unlabeled dataset, but over a different set of representative characters: „á“, „č“, „Ď“, „ç“, „Å“, „å“, „è“, and „Í“. If any of these characters were blank, undefined, or initialized with an error Unicode as in figure 3.2, the font was not selected. Upon manual assessment of the quality of this data, it was judged that no further Active Learning was necessary to improve the quality of this portion of the dataset with diacriticals.

In summary, fonts used in this method have been acquired online, with 222 462 used out of 272 849 unique fonts, including 7 089 fonts with selected diacritical marks (an acute accent ´, circle ˚, or caron ˇ on eight characters). The fonts have been downloaded from various sources, such as multiple unofficial datasets, Open Source libraries of fonts and font families, and official repositories of font-sharing websites. A font family is typically a group of related fonts which vary only in weight, orientation, width, etc., so in order to create a highly representative dataset, it is desirable to include fonts with similar variations. Downloaded fonts have been filtered with the deep net described earlier.

A Generative Adversarial Network was trained on this dataset to render any of the characters with and without diacriticals. While rendering data for training, the input was rendered as ordinary text with correct kerning and English letter statistics, by sampling phrases from a Harry Potter book. The GAN was simultaneously trained to generate diacritics, by using non-diacritics at the input with fonts containing diacritics, and a random accented character at the output. An outline of the trained GAN can be seen in figure 3.3, and further details on this standard process can be seen in the original publication [KHZ20].

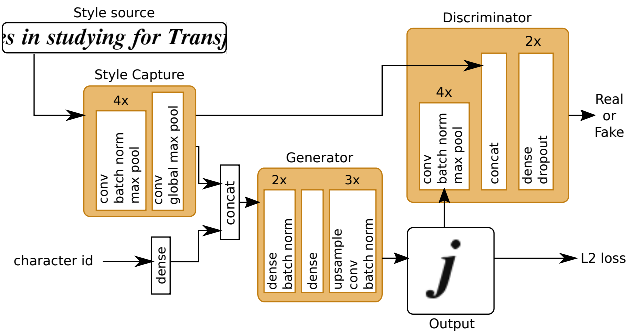


Figure 3.3: GAN structure, with sample inputs and outputs

3.1.1 User Study

A user study with 17 participants compared generated characters from state-of-the-art methods: VAE [18], ADV-VAE [33], and this work. The study was performed with three triplets of characters, as shown in

Figure 3.4. Each participant received 72 rows of triplets, printed on four sheets, and was asked to identify the different triplet. If the user fails to identify the generated triplet, the output of the method can be considered indistinguishable from the original font. Correct and incorrect user classifications are summed for each method, and results are presented in Table 3.1. The proposed method recreates fonts convincingly in 51% of cases, compared to 3% and 9% for the previous methods. According to the randomization permutation test, these results are highly significant (p<0.0001). Furthermore, tests show that VAE outperforms ADV-VAE with p-value 0.059.

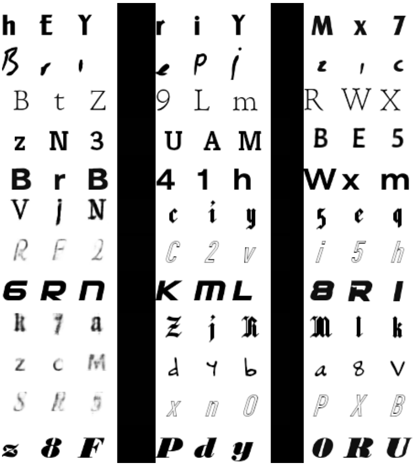


Figure 3.4: User Study Setup. Each row contains three triplets, two of which are ground truth, and one is generated by one of the three methods. Their order is randomized, except for the middle triplet, which is always the ground truth.

Using this approach, fonts may be extended to other alphabets and non-alphabetic languages for the benefit of billions of people whose native languages are not written in Latin alphabets. Unicode defines 136 900 characters [31], all of which can be generated in any font using

Method	success rate	adjusted success rate
VAE [18]	4.7%	9.4%
ADV-VAE [33]	1.6%	3.1%
Ours	25.6%	51%

Table 3.1: User Study Results. If the method produces indistinguishable outputs with respect to the ground truth, the user performs a random binary choice. This corresponds to a 100% method success rate for an ideal output, versus 50% measured in the experiment. The adjusted success rate is doubled accordingly.

this approach. Office suites such as Microsoft Office & Google Docs can benefit from incorporating such tools in the future.

A limitation of this work is the lack of kerning information. Currently, kerning is being done manually, so automated letter spacing is only possible for monospaced fonts, but this can be incorporated as an additional specialized task.

The hypothesis has been tested by comparing similar generative neural networks under similar conditions, but the network shown here has been trained with an improved larger and higher quality datasets thanks to the application of Active Learning. The method trained on the dataset produced via Active Learning has demonstrably outperformed the other, as shown in table 3.1.

3.2 Unsupervised Active Learning

Another practical application of Active Learning principles is Transfer Learning, enabling effective algorithmic annotation of unlabeled data. The Flick Faces HQ (FFHQ) dataset [16] has been successfully used in generating faces by training GANs, but the images lack annotation, which would provide useful information in guiding face generation via features. Due to this limitation, faces can only be generated by randomly sampling the latent space, or by selecting latent space points corresponding to known faces. However, it has been previously impossible to generate random faces with specific attributes.

The FFHQ dataset contains 70 000 unlabeled unique faces in high resolution, making it well suited for applications in graphics. See figure 3.5 for a random sample of these images. These images, even when annotated, would be impractical for training other tasks, such as gender recognition or orientation, because of the unnecessarily high definition and low sample count.

However, by combining state-of-the-art feature extractors with the high-resolution dataset, it is possible to create a dataset of labelled faces with useful information for guided generation of faces with specific features.

3.2.1 Transferring features

The dataset was created by running these pre-trained models to extract features: VGG-Face [25], Facenet [28], OpenFace [8], and DeepFace [29]. These pre-trained models detect faces and then produce features. These are geometrical features (landmarks and orientation), and well as categorical features (facial hair, emotion, eye colour, etc.). The 70 000 faces in were processed by these four models, and all but 528 very detected and annotated. The remaining faces are excluded from additional training in order to maintain the system's full autonomy. Transfer learning from pre-trained experts enables the creation of data labels by applying other algorithms, without human labels or supervision.

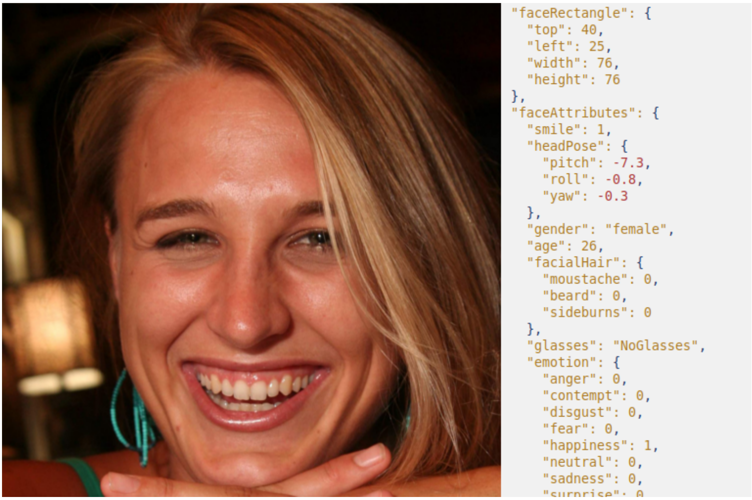
See figure 3.6 for a random annotated face. The resulting dataset, *FFHQ-features*, is available online[1].

3.2.2 Generative Faces with Features

The dataset was used to train a conditional Generative Adversarial Network. Unlike traditional GANs, which randomly sample the latent

[1] https://github.com/DCGM/ffhq-features-dataset

Figure 3.5: A sample from the Flickr Faces HQ dataset (FFHQ)[16]



```
"faceRectangle": {
  "top": 40,
  "left": 25,
  "width": 76,
  "height": 76
},
"faceAttributes": {
  "smile": 1,
  "headPose": {
    "pitch": -7.3,
    "roll": -0.8,
    "yaw": -0.3
  },
  "gender": "female",
  "age": 26,
  "facialHair": {
    "moustache": 0,
    "beard": 0,
    "sideburns": 0
  },
  "glasses": "NoGlasses",
  "emotion": {
    "anger": 0,
    "contempt": 0,
    "disgust": 0,
    "fear": 0,
    "happiness": 1,
    "neutral": 0,
    "sadness": 0,
    "surprise": 0
```

Figure 3.6: A random face from FFHQ dataset, with some extracted annotations in .json format

space to generate samples indistinguishable from training data with-

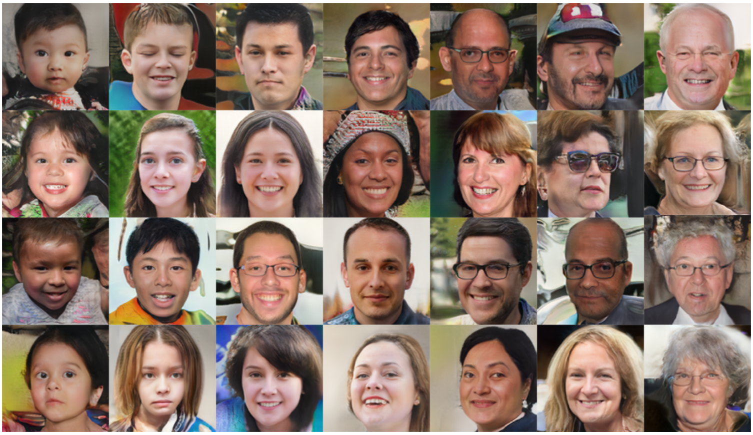


Figure 3.7: ProGAN-generated faces with age and gender restrictions. Rows alternate genders, columns hold incremental age groups [34]

out control over the features, conditional GANs are trained with an additional control vector, allowing them to set the desired properties of the output, given that this information was known during training.

Several architectures of cGANs were trained, with a number of different tunings, loss functions, and hyperparameters [34]. The features used were only age and gender, but the process can be applied to any categorical and continuous features present in the *FFHQ-features* dataset. Figure 3.7 shows randomly generated faces with gender and age control.

Instead of randomly sampling the latent space, and thus generating new faces, the network can also be used to generate the same face with different control features. In figure 3.8, the same random vector in latent space is rendered with different ages, resulting in the generated ageing process for a random, non-existent person.

By extending the use of Active Learning methods to fully algorithmic solutions via labelling with pre-trained networks, new results have been achieved. It was previously impossible to generate faces with spe-

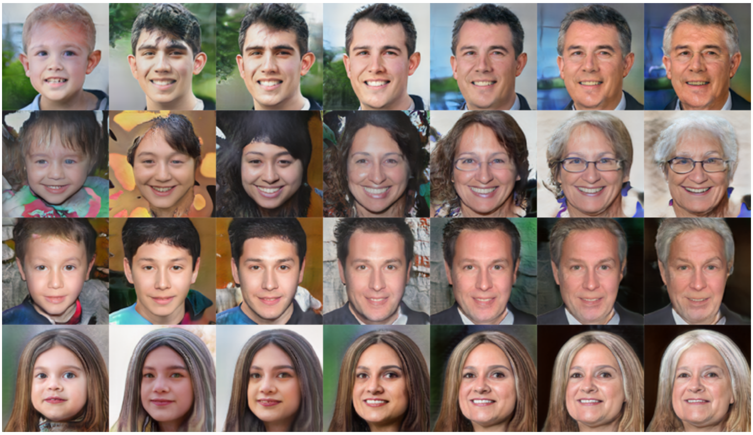


Figure 3.8: ProGAN-generated faces, with varying age parameters [34]

cific features due to the lack of such quality data, and by applying these techniques, the contemporary Machine Vision problem of generating faces has been quantitatively furthered.

3.3 Optimising user annotations

In this typical case of Adaptive Learning, a tagging system is presented such that it optimizes the annotation process with respect to two criteria: optimal adaptive recommendations based on prior actions, and an efficient interface for large-scale annotation. However, unlike Collaborative Filtering, this Active Learning use-case focuses on user-specific information, as opposed to preferences on globally known objects.

Most generally, users are presented with the option of creating arbitrary tags and aligning them to their own images. As these can be user-specific, language-specific, or location-specific, the information is not necessarily known for other users and other objects, and every tag has to be predicted online based on current tags.

By using Restricted Boltzmann Machines to provide labelling recommendations in a web-based user interface, the annotation of images via Active Learning has been experimentally tested in a user study. The objective of the tag suggestion methods is to allow Image-wise tagging (assign tags to an image) rather than Class-domain-wise tagging (assign images to a tag). These results demonstrate that large datasets with semantic labels (such as in TRECVID Semantic Indexing) can be annotated much more efficiently with the proposed approach than with current class-domain-wise methods, and produce higher quality data.

3.3.1 Local Tag Suggestion

A Restricted Boltzmann Machine is used to predict labels, by the encoding of the labels of surrounding tags and extracted features. Aside from the RBM suggestion method, tags are also suggested if they are positively annotated in nearby images in the gallery. A gallery is viewed as a chronological sequence, with images $\{I_i\}_{i=1}^{N}$. When generating suggestions for a given image I_i, each tag is given a weight ω, given by

$$\omega = \sum_{i=1}^{N} \frac{1}{log(|p-i|+1)} * has\_tag(I_i), \tag{3.1}$$

where

$$has\_tag(I_i) = \begin{cases} 1 & \text{if the tag is positively annotated on } I_i \\ -1 & \text{if the tag is negatively annotated on } I_i \\ 0 & \text{if the tag is not annotated on } I_i \end{cases}$$

The $\frac{1}{log(|p-i|+1)}$ term ensures that closer annotations have more weight on ω, and the $has\_tag(I_i)$ term ensures that positive annotations have positive weight, negative annotations negative weight, and all others are ignored. Tags are then ordered by their ω from highest to lowest. Any tags with $\omega > 0$ are then suggested, in this order.

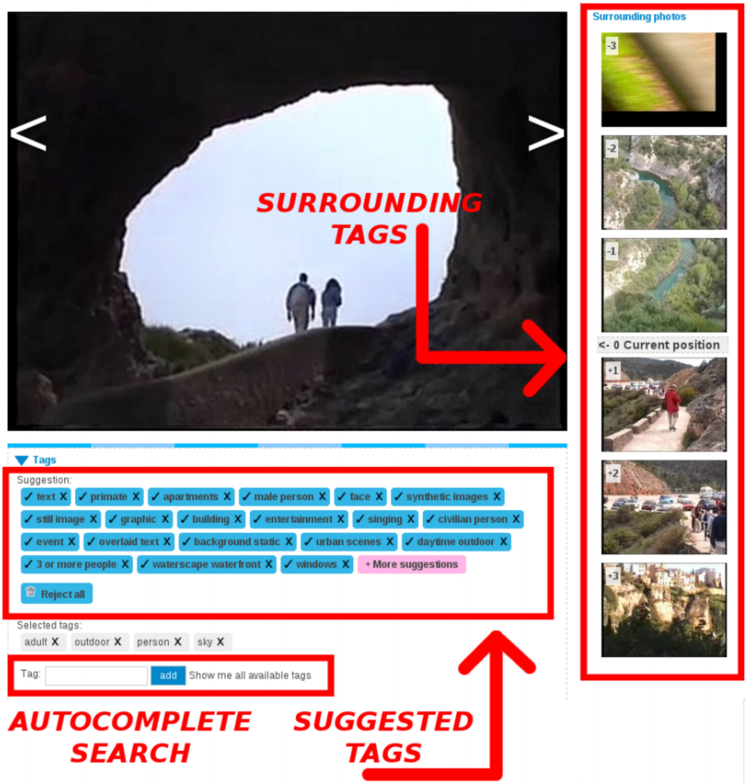


Figure 3.9: Typical view of the ITS web interface. Annotation option parts are outlined in red.

3.3.2 Integration of Suggestion and User Interface

When suggesting n tags, $\lfloor n/2 \rfloor$ are from the RBM model, $\lfloor n/2 \rfloor$ from local tag suggestion, and if n is odd, the remaining one is chosen with either method with equal probability. That ensures that when only one tag is being added, neither method is favoured.

When an image is loaded, 15 tags are chosen, and three annotating options are available to the user. As seen in Figure (picture of the Web), they are as follows:

1. Each of the 15 suggested tags is presented with a „check“ and a „cross“. When clicking check, the tag is added as positive annotation, and the cross adds negative annotation. When clicking either, the tag disappears from the suggestion list, and a new one is added at the end of the list.

2. The user can use an auto-completing text field, where any typed word or part of a word is matched with all occurrences in existing tags as a substring. For example, when typing „person“, the user is presented with „person“, „male person“, „female person“, and others. This ensures that when no information is given yet, the user can easily add information that's compatible with the current collection of tags in the database. When any of these is clicked, it gets added to the current suggestion, and the suggested tags are refreshed accordingly. Users are allowed to enter new tags which are not yet in the database; however, such tags are not immediately considered by the RBM model. It is more appropriate to add new tags to the RBM model when the number of positive annotations of such tags increases over a certain threshold in order to prevent saturating the model by rare or otherwise irrelevant tags.

3. Given the chronological sequence of images, three preceding and three succeeding images are shown on the right. When any of these is clicked, the positive tags that have been annotated on that image are copied over to the current image, and the suggested tags are refreshed accordingly.

The suggestion operation takes on average 0.1 seconds, making the system responsive and allowing quick interaction with the user. In case of sequential video frames, this interface allows users to seamlessly copy tags from previous images to the current one, either by copying tags from the three preceding and three succeeding images or by selecting the suggested local tags.

Another use scenario is the annotation of holiday photos with recurring themes, people, and elements. In the case of unusual images and tags that are not a priory likely, the RBM suggestions may not be accurate very useful at first; however, by providing one or several tags relevant to the image (e.g. by using the auto-completing text field) will make co-occurring tags likely to be suggested.

3.3.3 Experiments and Results

In order to identify the usability and usefulness of this system, two experiments with users were performed: testing with untrained individuals with minimal support, and testing with expert annotators for an extended period of time. In order to make the test replicable, only images and tags[2] from the TRECVID 2011 Semantic Indexing task[3] were used, and the feature to add new tags was disabled.

Besides the reproducibility of the experiments by others, there are several other advantages of using the TRECVID data. A part of the data is already annotated and can be used to learn the RBM tag-dependency model. Further, the dataset was annotated manually [5], which provides a baseline for comparison.

In addition to the user study, the ability of RBM to model dependencies among tags and the ability to estimate marginal tag probabilities by Gibbs sampling was tested on the TRECVID data. This experiment gives the objective information from the RBM suggestion system alone.

[2] Examples of the classes are Actor, Airplane Flying, Bicycling, Canoe, Doorway, Ground Vehicles, Stadium, Tennis, Armed Person, Door Opening, George Bush, Military Buildings, Researcher, Synthetic Images, Underwater and Violent Action.

[3] `http://www-nlpir.nist.gov/projects/tv2011/tv2011.html`

Testing by Untrained Users

Ten randomly selected technical university students were asked to use four different tag suggestion methods using this system, with as little training as possible. The four methods are:

1. **none** — no suggestion method
2. **RBM** — only Restricted Boltzmann Machine suggestion
3. **local** — only local tag suggestion (Section 3.3.1)
4. **RBM+local** — the combination of Restricted Bolzmann Machine and local tag suggestion, as presented in section 3.3.2

The methods were ordered randomly, and the user was not told which is which. After using each method, the user was asked to answer a questionnaire with questions regarding the rating and usability of the method, and data regarding the number of annotations created was stored.

According to the results (Figure 3.11), **RBM** and **RBM+local** suggestion methods allow significantly[4] faster annotation. There were no significant differences between **RBM** and **RBM+local**, nor between **none** and **local**. According to the questionnaire, method **none** is found by the users to be significantly[5] inferior to all the other methods in almost all aspects. No other significant differences were found, except that **RBM** and **RBM+local** received better marks in the ability to facilitate annotating more tags per image compared to **local**.

Testing by Expert Users

Three expert users were asked to use the combined tag suggestion method (Section 3.3.2). The users previously took part in TRECVID

[4] Using the paired t-test at the 10% significance level.

[5] Using the Mann-Whitney U test at the 10% significance level.

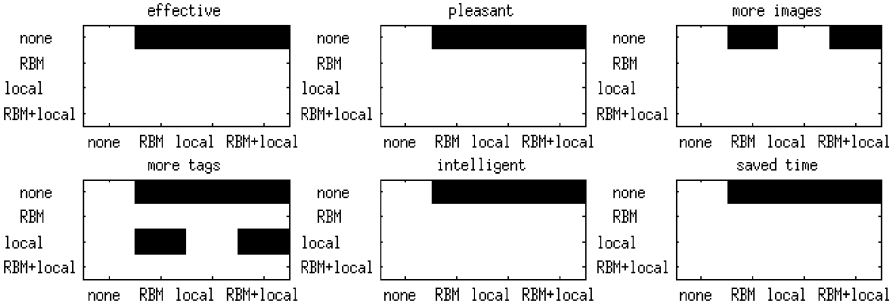


Figure 3.10: Black squares represent a significantly better outcome in the user evaluation, according to the questionnaire. The questions allowed a $1-5$ rating on effectiveness, pleasantness, amount of images, amount of tags per image, perceived method intelligence, and whether the method saved time.

2011 collaborative annotations [6] and had at least two hours experience with ITS. The users spent a total of three hours annotating randomly selected videos from the TRECVID dataset.

In this setting, the number of positive and negative annotations assigned per hour was 448 and 3085, respectively, averaging 13.1 positive annotations per image. The annotating speed compares very favorably to Class-domain-wise tagging annotation for which the authors of [6] expect 2 seconds per annotation; moreover, only 2.5% of the annotations in the TRECVID 2011 SIN [24] dataset are positive. When compared to the original distribution of tags obtained by the Active Learning method [6], the ITS tags have a heavier tail distribution for both positive (kurtosis 8.35 in TRECVID and 4.18 by ITS), and negative annotations (kurtosis 2.18 in TRECVID and 1.98 by ITS).

It has been shown that the presented method produces higher quality annotations in less time than comparable methods. Therefore, these results support the claim that Active Learning presents an improvement over approaches without it and that the creation of labelled datasets will benefit from the approach presented here.

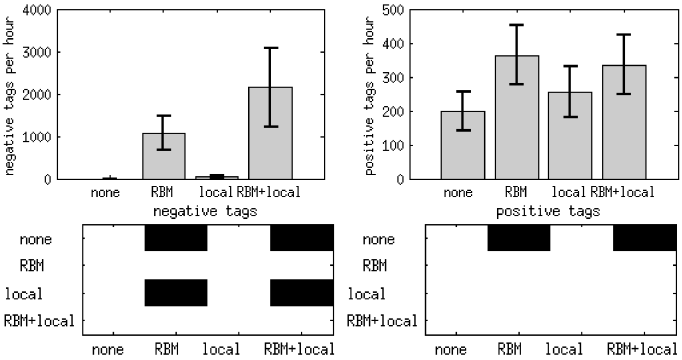


Figure 3.11: The top graphs show the mean number of tags assigned per hour with confidence intervals at 90% significance level. The bottom graphs show black squares where the column methods annotate significantly more tags per hour than the row methods.

3.4 Deep Learning on Small Datasets using Online Image Search

Learning image tags and object detectors is a core task of Computer Vision, and the large amount of data required to train every visual class is prohibitive. Therefore, by reformulating the problem in a Weakly Supervised PU learning setting, image categories can instead be trained from algorithmically preprocessed noisy online data. The following approach, the core contribution of this thesis, was presented at SCCG 2016 [KHZ16].

The proposed algorithm utilizes Google Image Search in a Hybrid Action Learning, where active learning with a weak algorithmic expert is used after an unsupervised initialization. Thousands of images are retrieved for any search string. The resulting set of images is weakly annotated, in the sense that numerous examples may be wrong or noisy. The data is stored statically for each given class, so this is not presented as a Online Learning problem but as an Incremental Learning problem.

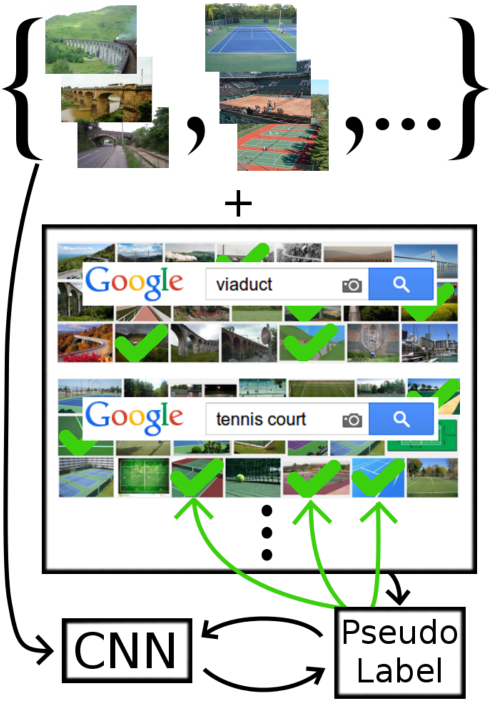


Figure 3.12: Pseudolabel selects useful additional images from an unreliable source, to help train a Deep Learning classifier

The proposed algorithm (Algorithm 1) is composed of an initial pre-training, a selection process, and a repeated weighted training step.

This section describes the data, the method, and the implementation.

In the original paper [21], pseudolabels are labels assigned during each epoch to any unlabeled images based on classifier responses. In the current setting, pseudolabels are weighted labels of the class used to query each image in online image search.

```
Data: labeled images, queried images for each class
Result: trained classifier
initial training of CNN with labeled images only;
while CNN not converged do
    for each queried image I do
        select whether to use I for training
    end
    train CNN with labelled and selected images
end
```

Algorithm 1: Proposed pseudolabel algorithm

Throughout this section, the following conventions are adopted: $\mathbf{X}$ is a set of images $\{X_1, X_2, X_3, \ldots\}$, $\mathbf{y}$ is a set of labels $\{y_1, y_2, y_3, \ldots\}$ where $y_n \in \{1, 2, \ldots C\}$. C denotes the number of categories. Training examples have the form $(\mathbf{X}, \mathbf{y})$. Every i model update iterations is referred to as one epoch, and a set of images and labels during the duration of epoch e is denoted $(\mathbf{X}_e, \mathbf{y}_e)$.

labeled images are divided into a training set and testing set: $(\mathbf{X}^{\text{train}}, \mathbf{y}^{\text{train}})$, $(\mathbf{X}^{\text{test}},$

In addition to the train and test sets, query images are retrieved from an online image search engine separately for each category. The queried images are denoted $(\mathbf{X}^{\text{query}}, \mathbf{y}^{\text{query}})$.

3.4.1 Training CNN

CNNs are trained by Stochastic Gradient Descent, where training images are propagated forward through the network in batches to produce outputs, for which error gradients are calculated. To complete an iteration, these are backpropagated to calculate loss gradients, which are used to update network weights. This process is repeated until convergence.

3.4.2 Pseudolabels with Query Images

The method described here relies on a different pseudolabel selection mechanism and a different pseudolabel weighting to the original approach [21]. When training with pseudolabel data, the CNN is trained as described in section 3.4.1. However, $\mathbf{X}^{\text{query}}$ images are repeatedly evaluated with the current network, and some are selected with pseudolabels $\mathbf{X}^{\text{pl}}$, for training.

At the beginning of training, $\mathbf{X}_0^{\text{pl}}$ is empty.

$$\mathbf{X}_0^{\text{pl}} = \emptyset \tag{3.2}$$

For the first i iterations (during epoch 0), the CNN is trained only with $(\mathbf{X}^{\text{train}}, \mathbf{y}^{\text{train}})$. Then, $\mathbf{X}_0^{\text{query}}$ is propagated forward through the CNN, to produce a set of vectors of beliefs for all labels $\mathbf{b}_0$ for every query image. These beliefs correspond to the normalized outputs of the last fully connected layer, before applying the last softmax layer.

Then, a randomized selection process chooses which predicted labels $\mathbf{y}^{\text{query}}$ will be trusted. Pseudolabel examples $\mathbf{X}_e^{\text{pl}}$ from the previous epoch are excluded.

$$(\mathbf{X}_{e+1}^{\text{pl}}, \mathbf{y}_{e+1}^{\text{pl}}) = selected(\mathbf{X}^{\text{query}} \setminus \mathbf{X}_e^{\text{pl}}, \mathbf{y}^{\text{query}}, \mathbf{b}_e) \tag{3.3}$$

The selection method proposed here is explained in section 3.4.3. The rest of $\mathbf{X}^{\text{query}} \setminus \mathbf{X}_e^{\text{pl}}$ is unused in this epoch.

This is the end of epoch 0. In each following epoch e, the CNN is trained with $\{(\mathbf{X}_e^{\text{pl}}, \mathbf{y}_e^{\text{pl}}), (\mathbf{X}^{\text{train}}, \mathbf{y}^{\text{train}})\}$. Section 3.4.4 discusses how $\mathbf{y}_e^{\text{pl}}$ can be weighted against $\mathbf{y}^{\text{train}}$ for better convergence stability.

3.4.3 Pseudolabel Selection

Each example image is chosen with probability:

$$\frac{(1-\lambda_c) * b_e}{2} \tag{3.4}$$

Where the accuracy λ_c for each class c on unlabeled data is the ratio of images classified as class c to the number of queried images in class c. By making the weak assumption that queried class accuracies across queried data are similar, class accuracies λ_c for the classifier are an indicator of training data and class complexity for each category.

The classifier belief b_e is the activation of the image for the queried class, as predicted by the network. By using the normalized belief in the $\mathbf{y}^{\text{query}}$ class, the selection favours images the classifier is more confident about, thus removing incorrect query images. This belief is normalized across network responses.

Classes with higher accuracy on the query dataset are given lower pseudolabel priority. This is accomplished with the $(1-\lambda_c)$ factor.

A number of factors affect the quantitative benefit of using pseudolabeled images: dataset belief, the accuracy of the selection method, the difference between datasets, selection variability over epochs, and randomization. This selection method balances these by selecting images in a randomized order, which depends on class accuracies and classifier belief for the correct class.

The last step is randomization. A portion of query images is randomly removed during selection. In these experiments, 50% were removed, and this was found to be beneficial. This is justified by a need to regularize across data when the CNN is trained.

3.4.4 Pseudolabel Weighting

Pseudolabels are likely to affect the classifier adversely when it hasn't yet reached a sufficient accuracy, just as the classifier would fail to train on raw query data. Self-training is prone to quickly converge to suboptimal solutions because the classifier assigns high confidence to wrong examples. How this is mitigated in this approach is explained below.

In the original pseudolabel paper [21], images from the training set have constant weights, and the pseudolabel losses are weighted by α, where α increases with time according to two hyperparameters.

Our experiments showed that this method is not more effective than setting $\alpha = 0$ until the network reaches near-top accuracy and then setting $\alpha = 1$. This method crucially relies on the network's ability to create a weak classifier from the training data alone, and it was found that this is the case with the previously published α tuning method

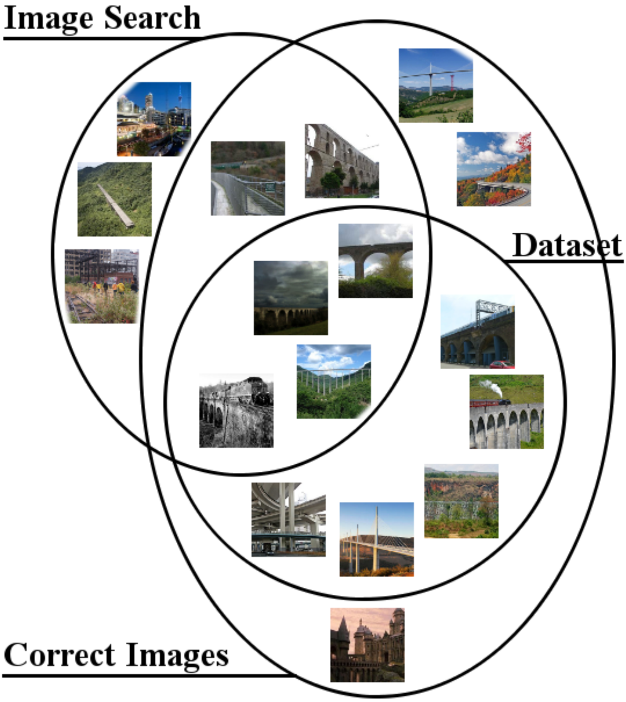


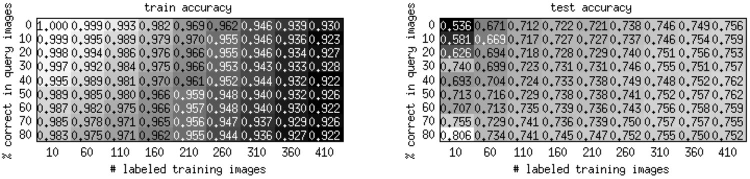
Figure 3.13: Example images of the viaduct class

as well. All shown results are achieved with this step function, thus demonstrating its usefulness.

This weighting method, albeit crude, simplifies hyperparameter tuning, and at the cost of a few epochs, achieves the same accuracy.

3.4.5 Dataset Belief

For an automatically retrieved set of images, a crucial piece of information for deciding whether to train using pseudolabels is the accuracy of the queried data. The unknown proportion of images which belong to the queried category is B, or dataset belief.



train accuracy

% correct in query images \ # labeled training images	10	60	110	160	210	260	310	360	410
0	1.000	0.999	0.993	0.982	0.969	0.962	0.946	0.939	0.930
10	0.999	0.995	0.989	0.979	0.970	0.955	0.946	0.936	0.923
20	0.998	0.994	0.986	0.976	0.966	0.955	0.946	0.934	0.927
30	0.997	0.992	0.984	0.975	0.966	0.953	0.943	0.933	0.928
40	0.995	0.989	0.981	0.970	0.961	0.952	0.944	0.932	0.922
50	0.989	0.985	0.980	0.966	0.959	0.948	0.940	0.932	0.926
60	0.987	0.982	0.975	0.966	0.957	0.948	0.940	0.930	0.922
70	0.985	0.978	0.971	0.965	0.956	0.947	0.937	0.929	0.926
80	0.983	0.975	0.971	0.962	0.955	0.944	0.936	0.927	0.922

test accuracy

% correct in query images \ # labeled training images	10	60	110	160	210	260	310	360	410
0	0.536	0.671	0.712	0.722	0.721	0.738	0.746	0.749	0.756
10	0.581	0.669	0.717	0.727	0.727	0.737	0.746	0.754	0.759
20	0.626	0.694	0.718	0.728	0.729	0.740	0.751	0.756	0.753
30	0.740	0.699	0.723	0.731	0.731	0.746	0.755	0.751	0.757
40	0.693	0.704	0.724	0.733	0.738	0.749	0.748	0.752	0.762
50	0.713	0.716	0.729	0.738	0.738	0.741	0.752	0.757	0.762
60	0.707	0.713	0.735	0.739	0.736	0.743	0.756	0.758	0.759
70	0.755	0.729	0.741	0.736	0.739	0.750	0.757	0.757	0.755
80	0.806	0.734	0.741	0.745	0.747	0.752	0.755	0.750	0.752

Figure 3.14: Train and test accuracies with varying correct query images, and varying train set sizes for each class

Query images can be wrong, misleading, and/or contain correctly and incorrectly labelled images from the training dataset, see Figure 3.13.

An imperfect selection must vary over epochs, in order to mitigate convergence to a non-median representation of the category.

3.4.6 Difference Between Datasets

If the training dataset and the images queried from online image search are the same, the method will not be of benefit. It is important that they are complementary, albeit with an overlap, and that they disagree to a degree. The disagreement creates jitter in the hyperspace between images where the classifier should not be divisive, and it supports convergence to a decision boundary elsewhere.

We found that selecting (X^{query}, y^{query}) which fully agrees with the current classifier does not boost classifier accuracy over not using pseudolabels at all. This is because despite bringing new information, the data doesn't create disagreement, and therefore no novelty. In these experiments, it was found that a certain degree of wrong and randomly labelled images helped the classifier to converge to higher accuracy over the test set. Adding this form of noise achieves regularisation.

3.4.7 Implementation

All images $\mathbf{X}^{\text{train}}, \mathbf{X}^{\text{test}}, \mathbf{X}^{\text{query}}$ were resized so that the smaller dimension is 227 pixels, and a central crop of 227 × 227 pixels is extracted. This has been shown to work better than other cropping methods [12], and the value 227 was chosen because this is the input size of the AlexNet network [19]. Preprocessing details are discussed and evaluated in [12].

The AlexNet [19] architecture was used and initialized with weights trained on the ImageNet dataset. The network was retrained by keeping all but the last fully connected layer locked, and by updating weights on the last layer.

The network was trained over 100 epochs of 500 iterations each with each combination of parameters. In a GPU-accelerated environment, such a network on the full SUN dataset with all query images converged in 2 to 5 hours.

The ratio of testing data to queried data accuracies is an indicator of the queried datasets accuracy or similarity. Assuming no constructive errors, such as those CNNs have been demonstrated to fall to when synthesizing examples [22], the number of correctly classified images is a lower bound on how many really belong into the category. A large difference between this number and the actual number (B), directly indicates how much further benefit the new data can have for training.

As shown in the right table 3.14, over test data, it can be seen that when the number of labelled images is small, the Active Learning approach using pseudolabels and a weak classified image retrieval system is of significant benefit. The accuracy can increase by as much as 25%, thus demonstrating that Active Learning benefits the critical Computer Vision task of learning image classifiers. In fact, the broad spectrum of classes and the small amount of data shows that this general approach can benefit many further tasks, well beyond the scope of this experiment.

3.5 Validating the Hypothesis

This chapter lists several experiments in which Active Learning has benefited contemporary Computer Vision, complementing existing algorithms via the judicious application of human labelling effort, or the use of pre-trained models for other, similar tasks.

Specifically, Active Learning has been shown to increase the quality of Generative Adversarial Networks for Font Capture by allowing the preparation of a larger and more representative dataset, it has enhanced the applicability of conditional GANs for generating faces by allowing the control of features, it has reduced the necessary time to manually annotate varied tags on images, and it has been shown to enable weak supervision to vastly improve the classification accuracy of image classifiers.

In terms of the symbolic formulation of the hypothesis, the method has shown that for various problems M and their associated loss functions L, there exists an Active Learning approach S which takes data X to produce parameters for the model which increase its accuracy over L. In practice, S can often benefit M even without the need for significant additional manual annotation, but by efficiently using Transfer Learning of existing algorithms as annotation experts.

The hypothesis, as stated in chapter 2, states that *Human and algorithmic expert annotation using Active Learning improves the accuracy of contemporary Computer Vision methods.* The work presented in this demonstrates this repeatedly for the human expert case, as well as for an algorithmic expert.

Therefore, this work validates the hypothesis formulated in the last chapter, with improvements by a significant margin to several contemporary Computer Vision problems.

CHAPTER 4

Conclusion

Machine Vision model quality is dependent on the versatility of the prior of the models used, on hyperparameters and parameter tuning, and the range and accuracy of data see during training. This work focuses on improving the accuracy of models by increasing data quality and quantity through Active Learning, validates the posed hypothesis, and demonstrates its benefits in a number of scenarios.

These main scientific contribution is the validation of the hypothesis, which stipulates that Active Learning benefits Computer Vision. This hypothesis is validated in two sets of differing scenarios: increasing the efficiency of manual labour for annotation, and utilizing Transfer Learning principles by applying pre-trained models to benefit a task. The applied contribution of this work is a series of experimental demonstrations of the hypothesis, and minor contributions are application-specific model improvements in Font Capture, One-shot-learning for image classification, and a tagging GUI to simplify annotation.

A system for human-assisted Image-wise tagging with suggestions was created, so that it could be used to obtain large semantically labelled datasets. The suggestion methods, as well as the annotating system itself, could be applied in the context of public media databases. The obtained annotations contain a higher percentage of positive examples of infrequent classes.

In another application, font capture benefited from Active Learning. Fonts are present in all forms of visual media, but working with them remains possible only for those with access to the type definitions. This work widens the possibilities for tools such as Photoshop and Google image translate, where recreating text in a given font is key. Automatically expanding the diacritical sets for existing fonts brings all fonts to a wider audience of hundreds of millions of users whose language includes diacritics.

Generative Adversarial Networks for generating faces have also seen an improvement thanks to Active Learning, by which new faces can be rendered with explicitly set features, such as gender and age. This benefit has come thanks to applying knowledge from other pre-trained models on existing data, showing that Active Learning is also beneficial in the case of algorithmic experts, rather than only with human annotators.

Similarly, an algorithmic expert in image retrieval was integrated into an extended pseudolabel training framework for CNN classifiers, demonstrating that Active Learning will push forward challenging tasks like image classification. This new method also does not require human supervision or annotation, bringing forth the possibility of extended applications by which Active Learning is applied seamlessly in Transfer Learning and Life-long learning tasks.

These specific tasks are some examples where my work has shown the benefit of Active Learning in increasing the quality of contemporary Computer Vision methods. While this validation is not a theoretical solution answering the hypothesis, this work will have demonstrated the general applicability of these principles and will enable a theoretical as well as a practical methodology to increase the quality of Computer Vision models at little cost.

In future work, it may be interesting to explore the question in a fashion systematic enough to allow automatic application, thus allowing the creation of an algorithm which searches for trainable models and existing datasets, and semi-automatically improves them using other known data and models. It will be particularly interesting to apply these principles to the other unsolved tasks where a large knowledge base can be drawn upon, such as theorem-proving, Computer Vision in video, and hyperspectral.

Glossary

Active Learning Process of selecting which data needs to get an expert label, either by a human or by another algorithm. 3–8, 12, 18, 23, 28, 38

Adaptive Learning Parameters are adjusted at runtime. 22

Association-rule Learning Discovering information about relationships. 7

Class-domain-wise tagging Assigning images to a tag. 23, 28

Collaborative Filtering Predicting preferences of users for objects given sparse preferences by other users. 22

Hybrid Action Learning Active Learning with unsupervised initialisation. 29

Image-wise tagging Assigning tags to an image. 23, 39

Incremental Learning Test-time input is used to improve the model. 29

One-shot Learning One or few examples while training. 8

Online Learning Training data is not statically available. 29

PU learning Positive and unlabeled data only. 29

Transfer Learning Adapting a pre-trained model. 6, 11, 18, 39

Weakly Supervised Learning on data with noisy, limited, or imprecise labels. 29

Zero-shot Learning Classifying to classes not seen during training. 8

Own work

[HKL+12] Michal Hradiš, Martin Kolář, Aleš Láník, Jiří Král, Pavel Zemčík, and Pavel Smrž. Annotating images with suggestions—user study of a tagging system. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 155–166. Springer, Berlin, Heidelberg, 2012.

[KCD15] Martin Kolář, Alan Chalmers, and Kurt Debattista. Repeatable texture sampling with interchangeable patches. *The Visual Computer*, 2015.

[KDC17] Martin Kolář, Kurt Debattista, and Alan Chalmers. A subjective evaluation of texture synthesis methods. In *Computer Graphics Forum*, volume 36, pages 189–198, 2017.

[KHZ16] Martin Kolář, Michal Hradiš, and Pavel Zemčík. Deep learning on small datasets using online image search. In *Proceedings of the 32nd Spring Conference on Computer Graphics*, pages 87–93, 2016.

[KHZ20] Martin Kolář, Michal Hradiš, and Pavel Zemčík. Capturing fonts in the wild. *The Visual Computer, under review*, 2020.

Bibliography

[1] A fontastic voyage: Generative fonts with adversarial networks. https://multithreaded.stitchfix.com/blog/2016/02/02/a-fontastic-voyage/, 2016.

[2] Font identifier. https://www.fontsquirrel.com/matcherator, 2017.

[3] Identifont. http://www.identifont.com, 2017.

[4] List of languages by number of native speakers. https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers, 2018.

[5] Stéphane Ayache and Georges Quénot. Evaluation of active learning strategies for video indexing. *Signal Processing: Image Communication*, 22(7-8):692–704, August 2007.

[6] Stéephane Ayache and Georges Quéenot. Video Corpus Annotation using Active Learning. In *European Conference on Information Retrieval (ECIR)*, pages 187–198, Glasgow, Scotland, March 2008.

[7] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018.

[8] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. Openface 2.0: Facial behavior analysis toolkit. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 59–66. IEEE, 2018.

[9] Shumeet Baluja. Learning typographic style: from discrimination to synthesis. *Machine Vision and Applications*, 28(5-6):551–568, 2017.

[10] Erik Bernhardsson. Analyzing 50k fonts using deep neural networks. https://erikbern.com/2016/01/21/analyzing-50k-fonts-using-deep-neural-networks.html, 2016.

[11] Neill DF Campbell and Jan Kautz. Learning a manifold of fonts. *ACM Transactions on Graphics (TOG)*, 33(4):91, 2014.

[12] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[13] Taylor Childers, Jessica Griscti, and Liberty Leben. 25 systems for classifying typography: A study in naming frequency. *PJIM Parsons Journal for Information Mapping*, 5(1), 2013.

[14] Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European conference on computer vision*, pages 71–84. Springer, 2010.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database.

In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[17] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[18] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] DH Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on Challenges in Representation Learning*, 2013.

[22] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CVPR*, 2015.

[23] Peter O'Donovan, Jānis Lībeks, Aseem Agarwala, and Aaron Hertzmann. Exploratory font selection using crowdsourced attributes. *ACM Transactions on Graphics (TOG)*, 33(4):92, 2014.

[24] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, and Georges Quéenot. TRECVID 2011 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *Proceedings of TRECVID 2011*. NIST, USA, 2011.

[25] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 41.1–41.12. BMVA Press, 2015.

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. IEEE, Jun 2015.

[29] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708. IEEE, 2014.

[30] Joshua B Tenenbaum and William T Freeman. Separating style and content. In *Advances in neural information processing systems*, pages 662–668, 1997.

[31] The Unicode Consortium. The Unicode Standard. Technical Report Version 10.0.0, Unicode Consortium, 2011.

[32] Antonio Torralba, Rob Fergus, and William T Freeman. 80 Million Tiny Images: a Large Data Set for Nonparametric Object and Scene Recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–70, November 2008.

[33] Paul Upchurch, Noah Snavely, and Kavita Bala. From a to z: Supervised transfer of style and content using deep neural network generators. *arXiv*, pages arXiv–1603, 2016.

[34] Tomáš Venkrbec. Generating faces with conditional generative adversarial networks. *VUT FIT Bachelor Thesis*, 2020.

[35] Jenny Yuen, Bryan Russell, Ce Liu, and Antonio Torralba. LabelMe video : Building a Video Database with Human Annotations. *Event (London)*, pages 1–8.