The purpose of this document is to collect information relevant to the development from different sources.

I have no authorship on the contents of this document, and do not intend to claim any by compiling the different information in this place.

The source of each piece of information is appropriately cited when relevant.

Relevant topics (index):

* The Weather Classification problem
  + Description
  + Applications
  + Approaches
* Convolutional Neural Networks
  + Inception
  + History
  + Traditional Architecture
  + Future work
  + Implementations
    - AlexNet
      * <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
    - GoogLeNet
      * <https://arxiv.org/pdf/1409.4842v1.pdf>
    - ResNet
      * <https://arxiv.org/pdf/1512.03385v1.pdf>
    - Places
      * <http://places.csail.mit.edu/places_NIPS14.pdf>
    - VGGNet
      * <https://arxiv.org/pdf/1409.1556v6.pdf>
    - VGGCNN F/M/S
      * <http://www.robots.ox.ac.uk/~vgg/publications/2014/Chatfield14/chatfield14.pdf>
* Caffe framework for Deep Learning
* Superpixels

# The Weather Classification problem

## Description

“Vision based driver assistance systems (DAS) are currently designed to perform under good-natured weather conditions. Unfortunately, limited visibility often occurs in daily life (e.g. heavy rain or fog). As this strongly affects the accuracy or even the general function of vision systems, the actual weather condition is a valuable information for assistance systems. Based on the results of weather classification, specialized approaches for each class can be invoked to improve cognition. This will form a key factor to expand the application of DAS from selected environmental conditions to an overall approach.” [1]

“The weather affects our daily lives in many ways, from solar technologies, outdoor sporting events, to the sort of clothes we wear and whether to stay indoors or not on weekend.

While current accurate weather detection technologies rely on expensive sensors, for centuries weather observing tools consisted of the human eye (and various human senses as well). If we can exploit existing surveillance cameras, which are found almost everywhere, it may be possible to turn weather observing and detection into a powerful and cost-effective computer vision application.” [2]

“The weather conditions not only strongly influence us in our daily lives [1] through the solar energy system and outdoor sporting events as examples, but also affects the functionality of many visual systems including outdoor video surveillance and vehicle assistant driving systems [2, 3] (by heavy rain, haze, etc.). It is no doubt that, judging the weather conditions by a single image, also known as weather classification task, plays a vital role in many visual and weather systems. Nowadays, the weather classification task is commonly accomplished by the human vision or expensive sensors. Since weather condition is local to an area, lack of the required human resources and/or the expensive sensors limits the avail- ability of local measurement of the weather condition. Recently, researchers argued that computer vision techniques could be developed to accurately classify weather conditions through images, which might save expensive human and instrumental resources (i.e., sensors) since economical surveillance cameras are ubiquitous and would be sufficient to accomplish weather classification. In this paper, we refer to weather classification from images as the task of predicting the class of the weather given an image (e.g., cloudy, sunny, etc.).” [3]

## Applications

## Approaches

### Multi-Class Weather Classification

“Different from the works above, we propose a method for classifying multi-class weather from single images which is based on multiple weather features and multiple kernel learning. Implementation of this idea, however, entails substantial challenges. First, it is difficult to find the suitable features to discriminate different weather. Second, the features might be heterogeneous and the feature vectors are high-dimensional. Aiming at the above challenges, firstly, we extract multiple features to represent different weather.” [4]

### Two-Class Weather Classification

“Our first technical contribution consists of the design and implementation of the various weather cues which are used to form the *weather feature*. These everyday weather cues (such as sky, shadow, reflection, contrast and haze) are what human are still using for weather observing – a hazy or grayish sky characterizes a cloudy day, hard shadow cast on ground indicates a sunny day, as illustrated in Figure 3(a). Conversely, in the absence of any weather cues, we ourselves would lower the confidence to correctly label weather, as shown in Figure 3(b).

Given the weather feature, the next question is how to properly learn the classifier. The main issue is that not all of the weather cues are available in an image (e.g., not every outdoor image has a sky region), which is problematic to a discriminative training process adopted by traditional classifiers, such as SVM. To address this problem, our second technical contribution consists of a collaborative learning framework using homogeneous voters: we group outdoor images into clusters where images in the same cluster are similar in terms of the weather cues. This allows us to build classifiers in a conventional way thanks to the homogeneity in each cluster. The final labeling is the weighted voting result of the cluster classifier outputs. The cluster closer to the testing image is given a larger weight. As will be explained in the following, homogeneous voters are learned together in synergy under a unified optimization framework.

Despite the absence of representative work on image- based weather labeling, we perform quantitative comparison with a few common baselines including SVM, Adaboost [19, 22], and weather-related prior methods [9, 21, 16]. Perspectives on related work will be put into context when they are described.

Our final contribution consists of a 10K weather image dataset properly selected and annotated. This is used to evaluate our learning and labeling strategy.

Our learning strategy is to partition training images into disjoint clusters of homogeneous voters. Given a test image, voters closer to it are given more weights for correctly finding the weather label.” [5]

“Although the previous works provide interesting solu- tions for weather classification, the performances of these approaches are unappealing. As far as we know, the best nor- malized classification accuracy achieved in the challenging weather image dataset, which consists of 10K images, is only 53.1% (the regular accuracy is 76.5%) [1]. Figure 1 (left) shows a challenging image with sunny weather condition. It was also reported in [1] that the histogram of mean light- ness of sunny and cloudy images substantially overlap, which makes the dataset very challenging. We attribute the low performance of [1] mainly to the engineered image features adopted in this method. Compared to the typical image clas- sification task, weather classification from images is affected by various factors, e.g., illumination, reflection, scene and shadow. These factors are highly coupled with each other and therefore the categorization manifold is highly nonlinear. Al- though the previous engineered approaches can satisfy some desirable properties and mitigate some undesirable properties from these factors, they cannot well capture such nonlinearity of the categorization manifold, which makes discrimination between weather classes a hard problem.” [3]

### Multi-Class Weather Classification On Single Images

“we propose a method for classifying multi-class weather from single images which is based on multiple weather features and multiple kernel learn- ing. Implementation of this idea, however, entails substantial challenges. First, it is difficult to find the suitable features to discriminate different weather. Second, the features might be heterogeneous and the feature vectors are high-dimensional. Aiming at the above challenges, firstly, we extract multiple features to represent different weather. For example, the sky and shadow features can indicate the sunny weather, the haze feature can indicate the haze weather, the HOG based tem- plate matching feature can indicate the rainy weather, the snowflake noise feature can indicate the snowy feature, and some global features like contrast and saturation are used to distinguish multi-class weather. Secondly, we utilize multiple kernel learning to learn an adaptive classifier to fuse these heterogeneous and complementary features effectively. Compared with the traditional classification methods (such as SVM, Adaboost), multiple kernel learning is beneficial to learn an adaptive classifier because it can choose the best combination of kernels.

The contribution of this paper lies in four aspects:

* To the best of our knowledge, the proposed method is the first to focus on the problem of multi-class weather classification from single images in any scenario.
* We propose two methods for detecting rain and snow from single images respectively.
* We improve some existing algorithms for obtaining the weather features and take multiple kernel learning for multi-class classification.
* We collect an outdoor image set that contains 20K im- ages called MWI (Multi-class Weather Image) set.” [4]

## Approaches with CNNs

### Weather Classification with Deep Convolutional Neural Networks

“Motivated by the remarkable successes of Convolutional Neural Networks (CNNs) in computer vision and machine learning [5, 6, 7, 8, 9], we adopt CNNs to solve the weather classification task. There are three reasons for us to choose this technique: The CNN is a neural network model which captures nonlinear mapping between the different spaces, e.g.,. feature space and label space; Deep CNN has demonstrated the powerful discriminating power in extensive image representation and classification tasks; CNNs are simple and explicit end-to-end convolutional architectures, which can simplify the weather classification, without the need for engineered features (e.g. HOG [10], GIST [11]).

Most CNN works are designed for addressing the object recognition and detection tasks [5, 6, 8, 9]. However, weather classification is quite different to these issues. It is more sensitive to factors, such as lighting condition and the status of sky and shadows, rather than object-related information, such as shape and texture. This paper focuses on studying the feature spaces introduced by the different layers of CNN in the weather classification task. There are three main questions that we aim to answer:

* How good is the representation at different layers of a pre-trained CNN for addressing the weather classification problem?
* How fine-tuning of a pre-trained CNN optimized for weather classification dataset will affect the representation at each layer of the network?
* How spatial coherence is important in CNN-based weather classification?

We conducted several experiments to address all these questions over different layers of the CNNs for the weather problem. Adopting CNNs, we concluded our work by significantly outperforming the state-of-the-art by 54.8%.” [3]

# Convolutional Neural Networks

## Inception/What are CNNs?

“Conventional machine-learning techniques were limited in their ability to process natural data in their raw form. For decades, constructing a pattern-recognition or machine-learning system required careful engineering and considerable domain expertise to design a feature extractor that transformed the raw data (such as the pixel values of an image) into a suitable internal representation or feature vector from which the learning subsystem, often a classifier, could detect or classify patterns in the input.

Representation learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. Deep-learning methods are representation-learning methods with multiple levels of representa- tion, obtained by composing simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level. With the composition of enough such transformations, very complex functions can be learned. For classification tasks, higher layers of representation amplify aspects of the input that are important for discrimination and suppress irrelevant variations.

The key aspect of deep learning is that these layers of features are not designed by human engineers: they are learned from data using a general-purpose learning procedure.

A deep-learning architecture is a multilayer stack of simple mod- ules, all (or most) of which are subject to learning, and many of which compute non-linear input–output mappings. Each module in the stack transforms its input to increase both the selectivity and the invariance of the representation. With multiple non-linear layers, say a depth of 5 to 20, a system can implement extremely intricate func- tions of its inputs that are simultaneously sensitive to minute details — distinguishing Samoyeds from white wolves — and insensitive to large irrelevant variations such as the background, pose, lighting and surrounding objects.

” [5]

## Implementations

#### Detection of Human Rights Violations in Images: Can Convolutional Neural Networks help?

“This work is made possible by recent progress in Convolutional Neural Networks (CNNs) [LeCun et al., 1989], which has changed the landscape for well-studied computer vision tasks, such as image classification and object detection [Wang et al., 2010, Huang et al., 2011], by comprehensively outperforming the initial handcrafted approaches [Donahue et al., 2014, Sharif Razavian et al., 2014, Sermanet et al., 2013].

As part of our tests, we delve into the latest, top-performing pre-trained deep convolutional models, allowing a fair, unbiased comparison on a common ground; something that has been largely missing so far in the literature.

While large-scale datasets combined with CNNs have been key to recent advances in computer vision and machine learning applications.

which was constructed by utilizing images collected by non-vision/machine learning researchers, by querying Flickr with a number of related keywords, including the class name, synonyms and scenes or situations where the class is likely to appear.

For decades, traditional machine learning systems demanded accurate engineering and significant domain expertise in order to design a feature extractor capable of converting raw data (such as the pixel values of an image) into a convenient internal representation or feature vector from which a classifier could classify or detect patterns in the input.

are driving advances at a dramatic pace in the computer vision field after enjoying a great success in large-scale image recognition and ob- ject detection tasks [Krizhevsky et al., 2012,Sermanet et al., 2013, Simonyan and Zisserman, 2014a, Tomp- son et al., 2015, Taigman et al., 2014, LeCun et al., 2015].

The layers of features are not manu- ally hand-crafted, but are learned from data using a generic-purpose learning scheme.

In the last few years vision tasks became feasible due to high-performance computing systems such as GPUs, extensive public image repos- itories [Deng et al., 2009], a new regularisation tech- nique called *dropout* [Srivastava et al., 2014] which prevents deep learning systems from overfitting, rec- tified linear units (ReLU) [Nair and Hinton, 2010], softmax layer and techniques able to generate more training examples by deforming the existing ones.

Since [Krizhevsky et al., 2012] first used an eight layer CNN (also known as AlexNet) trained on ImageNet to perform 1000-way object classification, a number of other works have used deep convolutional networks (ConvNets) to elevate image classification further [Simonyan and Zisserman, 2014b, He et al., 2015a, Szegedy et al., 2015, He et al., 2015b, Chat- field et al., 2014]. [Simonyan and Zisserman, 2014b] use a very deep CNN (also known as VGGNet) with up to 19 weight layers for large-scale image clas- sification. They demonstrated that a substantially increased depth of a conventional ConvNet [LeCun et al., 1989, Krizhevsky et al., 2012] can result in state-of-the-art performance on the ImageNet chal- lenge dataset [Deng et al., 2009]. They also per- form localization for the same challenge by training a very deep ConvNet to predict the bounding box location instead of the class scores at the last fully connected layer. Another deep network architecture that has been recently used to great success is the GoogLeNet model of [Szegedy et al., 2015] where an inception layer is composed of a shortcut branch and a few deeper branches in order to improve utilization of the computing resources inside the network. The two main ideas of that architecture are: (i) to create a multi-scale architecture capable of mirroring correlation structure in images and (ii) dimensional reduction and projections to keep their representation sparse along each spatial scale. Most recently [He et al., 2015a] announced the even deeper residual network (also known as ResNet), featured 152 layers, which has considerably improved the state-of-the-art performance of ImageNet [Deng et al., 2009] classification and object detection on PASCAL [Everingham et al., 2010]. Residual networks are inspired by the observation that neural networks lean towards gaining higher training errors as the depth of the network increases to very large values. The authors argue that although the network gains more parameters by increasing its depth, the network becomes inferior at function approximation because of the gradients and training signals loss when they are propagated through numerous layers. Therefore, they give convincing theoretical and practical evidence that residual connections (re- formulated layers for learning residual functions with reference to the layer input) are inherently necessary

for training very deep convolutional models. Outside of the aforementioned top-performing networks, other works worth mentioning are: [Chat- field et al., 2014] where a rigorous evaluation study on different CNN architectures for the task of object recognition was conducted and [Zhou et al., 2014] where a brand-new scene-centric database called Places was introduced and established state-of-the- art results on different scene recognition tasks, by learning deep representations from their extensive database. Despite these impressive results, human rights advocacy is one of the high profile domains which remain broadly missing from the curated list of problems which were benefited from the continuing growth of deep convolutional networks. We build on this body of work in deep learning to solve the untrodden problem of recognising human rights violations utilising digital images.

use a pre-trained model and then use the ConvNet as a fixed feature extractor for the task of interest.

taking a pre-trained CNN, replacing the fully-connected layers (and potentially the last convolutional layer), and consider the rest of the ConvNet as a fixed feature extractor for the relevant dataset. By freezing the weights of the convolutional layers, the deep ConvNet can still extract general image features such as edges, while the fully connected layers can take this information and use it to classify the data in a way that is applicable to the problem.

every block is fixed except the feature ex- tractor as different deep convolutional networks are plugged in, one at a time, to compare their perfor- mance utilizing the mean average precision (mAP) metric.

Given a training dataset *Tr* consisting of m human rights violation categories, a test dataset *Ts* comprising unseen images of the categories given in *Tr*, and a set of n pre-trained CNN architectures (*C*1 ,...*Cn* ), the pipeline operates as follows: The training dataset *Tr* is used as input to the first CNN architecture *C*1. The output of *C*1, as described above, is then utilized to train m SVM classifiers. Once trained, the test dataset *Ts* is employed to assess the performance of the pipeline using mAP. The training and testing procedures are then repeated after replacing *C*1 with the second CNN architecture *C*2 to evaluate the performance of the human rights violation recognition pipeline. For a set of n pre-trained CNN architectures, the training and testing processes are repeated n times.

for all n CNN architectures, the differences in the performance of the classification pipeline can be attributed to the specific CNN architectures used.

To ensure a fair comparison, all the standardised CNN models used in our experiments are based on the opensource Caffe framework [Jia et al., 2014] and are pre-trained on 1000 ImageNet [Deng et al., 2009] classes with the exception of Places CNN [Zhou et al., 2014] which was trained on 205 scenes categories of Places database. For the majority of the networks, the dimensionality of the last hidden layer (FC7) leads to a 4096x1 dimensional image representation. Since the GoogLeNet [Szegedy et al., 2015] and the ResNet [He et al., 2015a] architectures do not utilise fully connected layers at the end of their net- works, the last hidden layers before average pooling at the top of the ConvNet are exploited with 1024x7x7 and 2048x7x7 feature maps respectively, to counter- balance the behaviour of the pool layers, which pro- vide downsampling regarding the spatial dimensions of the input.

##### Evaluation

The evaluation process is divided into two different sets of scenarios, each one making use of an explicit split of images between the training and testing samples of the pipeline. For the first scenario, a split of 70/30 was utilised, while for the second scenario the split was adjusted to 50/50 for training and testing images respectively. Additionally, three distinct series of tests were conducted for each scenario, each and every one assembled with a completely arbitrary shift of the entire image set for every category of the HRUN dataset. This approach ensures an unbiased comparison with a rather limited dataset like HRUN at present. The compound results of all three tests are given in Table 2 and Table 3 and analysed below.

Such weaker performance occurs primarily because of the limited dataset size, whereby learning millions of parameters of those very deep convolutional networks is usually impractical and may lead to over-fitting.

Furthermore, it is clear that by utilising the 50/50 split of images in the course of scenario 2, there is a considerable boost in performance of the human rights violations recognition pipeline as compared to the first scenario when a split of 70/30 was employed for training and testing images respectively.

##### Conclusions

The following conclusions have derived: Digital images that can be rated as appropriate for human rights monitoring purposes are rare and characterising them requires great effort, expertise and vast time. Utilising transfer learning for the task of recognising human rights violations can provide very strong results by employing a straightforward combination of deep representations and a linear SVM. Deep convolutional neural networks are constructed to benefit and learn from massive amounts of data. For this reason and in order to obtain even higher quality recognition results, training a deep convolutional network from scratch on an expanded version of the HRUN dataset is likely to further improve results. Inspired by the high-standard characteristics of legal evidence, in the future we would like to have the means to clarify three different questions set by every human rights monitoring mechanism: *what*, *who* and *how*, and expand our dataset to a wider range of categories in order to include them. We also presume that further analysis of joint object recognition and scene understanding will be beneficial and lead to improvements in both tasks for human rights violations understanding.” [6]

# Superpixels

## What are Superpixels?

“Superpixels provide a convenient primitive from which to compute local im- age features. They capture redundancy in the image [1] and greatly reduce the complexity of subsequent image processing tasks. They have proved increasingly useful for applications such as depth estimation [2], image segmentation [3, 4], skeletonization [5], body model estimation [6], and object localization [7].

For superpixels to be useful they must be fast, easy to use, and produce high quality segmentations. Unfortunately, most state-of-the-art superpixel methods do not meet all these requirements. As we will demonstrate, they often suffer from a high computational cost, poor quality segmentation, inconsistent size and shape, or contain multiple difficult-to-tune parameters. ” [7]

## Simple Linear Iterative Clustering (SLIC) Algorithm

“The algorithm we propose, simple linear iterative clustering (SLIC) performs a local clustering of pixels in the 5-D space defined by the L,a,b values of the CIELAB color space and the x,y pixel coordinates. A novel distance measure enforces compactness and regularity in the superpixel shapes, and seamlessly accomodates grayscale as well as color images. SLIC is simple to implement and easily applied in practice – the only parameter specifies the desired number of superpixels. Experiments on the Berkeley benchmark dataset [11] show that SLIC is significantly more efficient than competing methods, while producing segmentations of similar or better quality as measured by standard boundary recall and under-segmentation error measures.

For many vision tasks, compact and highly uniform superpixels that respect image boundaries, such as those generated by SLIC in Fig. 1, are desirable.

Loose or irregular superpixels can degrade the performance. Local features such as SIFT extracted from the image at superpixel locations become less meaningful and discriminative if the superpixels are loose or irregular, and learning statistics over cliques of two or more superpixels can be unreliable.” [7]

# Works Cited

|  |  |
| --- | --- |
| [1] | M. Roser and F. Moosmann, "Classification of Weather Situations on Single Color Images," in *2008 IEEE Intelligent Vehicles Symposium*, Eindhoven, 2008. |
| [2] | C. Lu, D. Lin, J. Jia and C.-K. Tang, "Two-Class Weather Classification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, 2014. |
| [3] | M. Elhoseiny, S. Huang and A. Elgammal, "Weather classification with deep convolutional neural networks," in *IEEE International Conference on Image Processing, ICIP 2015*, Quebec City, 2015. |
| [4] | Z. Zhang and H. Ma, "Multi-class weather classification on single images," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015. |
| [5] | Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature,* vol. 521, pp. 436-444, 28 May 2015. |
| [6] | G. Kalliatakis, S. Ehsan, F. Maria , A. Leonardis, J. Gall and K. D. McDonald-Maier, "Detection of Human Rights Violations in Images: Can Convolutional Neural Networks help?," in *12th International Conference on Computer Vision Theory and Applications (VISAPP 2017)*, Portugal, 2017. |
| [7] | R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels," EPFL, Lausanne, 2010. |
| [8] | Z. Zhu, L. Zhuo, P. Qu and K. Zhou, "Extreme Weather Recognition using Convolutional Neural Networks," in *2016 IEEE International Symposium on Multimedia*, San Jose, 2016. |
| [9] | A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *26th Annual Conference on Neural Information Processing Systems 2012*, Lake Tahoe, 2012. |
| [10] | C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," in *2015 7th International Conference on Games & Virtual Worlds for Serious Applications*, Boston, 2015. |
| [11] | K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016. |
| [12] | B. Zhou, A. Lapedriza, J. Xiao, A. Torralba and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *Advances in Neural Information Processing Systems 27 - 28th Annual Conference on Neural Information Processing Systems 2014*, Montreal, 2014. |
| [13] | K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, San Diego, 2015. |
| [14] | K. Chatfield, K. Simonyan, A. Vebaldi and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," in *Proceedings of the British Machine Vision Conference 2014*, Nottingham, 2014. |
| [15] | J. Yan, Y. Yu, X. Zhu, Z. Lei and S. Z. Li, "Object Detection by Labeling Superpixels," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 2015. |