# BLUR ASSESSMENT IN SIGN LANGUAGE VIDEOS

by
Giray Naim Eryılmaz
B.S., Computer Engineering, Boğaziçi University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2022

## ABSTRACT


## BLUR ASSESSMENT IN SIGN LANGUAGE VIDEOS

Blur impairs the sharpness of visual features and the clarity of details. It may sometimes be desired for artistic effect. However, in general, it is regarded as a defect. There are different problems studied about blur, such as blur detection, segmentation, estimation, and deblurring, but despite its abundance in visual media such as photographs and videos, there is limited annotated data about blur. This lack of data inhibits the usage of deep learning models because they require a lot of annotated data. Annotating that much data is expensive and cumbersome. In this thesis, we investigate blur-vs-sharp classification using deep learning, also we experiment with weak supervision as a remedy against the lack of data for blur assessment and localization. We compare our results with the classical approaches found in the literature. We use the data we annotated from four different datasets, three of which are sign language datasets and the other one is an action recognition dataset. We focus our research on sign language videos where motion blur is frequently encountered. Sign languages are the primary communication method of Deaf community and for that reason sign language recognition (SLR) is an important task. Determining the intensity of blur and its location may be beneficial for SLR research.

# ÖZET


## İŞARET DİLİ VİDEOLARINDA BULANIKLIK DEĞERLENDİRMESİ

Bulanıklık görsellerdeki hatları ve detayları dağıtır, dolayısıyla netliği düşürür. Bazen sanatsal değeri için istense de genelde bir noksan olarak görülür. Literatürde bulanık tespiti, segmentasyonu, ölçümü ve bulanıklığın giderilmesi gibi bulanık ile alakalı çeşitli problemler bulunur. Videolar ve fotoğraflar gibi görsel medyalarda bolca rastlanmasına karşın bulanıklık ile alakalı işaretlenmiş veri oldukça azdır. Bu durum eğitilmeleri için çok miktarda veri gerektiren derin öğrenme modellerinin kullanım olanaklarını kısıtlamaktadır. Bu modelleri kullanmayı mümkün hale getirecek kadar veri işaretlemek oldukça maliyetlidir. Biz bu tez çalışmasında, bulanıklık içeren/içermeyen görsellerin sınıflandırılması problemi üzerine çalışıyoruz. Ayrıca bulanıklığın derecesinin belirlenmesi ve görselde yerinin tespit edilmesi problemlerinde veri eksikliğine bir çare olarak zayıf gözetimin kullanımı ile ilgili araştırma ve deneylerde bulunuyoruz. Elde ettiğimiz sonuçları literatürde bulunan klasik metodlarınkilerle karşılaştırıyoruz. Üç tanesi işaret dili videolarından oluşan ve bir diğeri eylem tanıma videolarından meydana gelen toplam dört veri setinden alıp işaretlediğimiz kareleri bu çalışmalarımızda kullanıyoruz. Çalışmalarımızı bulanıklığın sık sık gözlendiği işaret dili videolarına yönlendirdik. İşaret dili, duyma engellilerin esas iletişim yöntemidir ve işaret dili tanıma çalışmaları bu sebeple büyük önem arz ediyor. Bulanıklığın veride bulunup bulunmadığının, eğer bulunuyorsa ne miktarda ve nerede bulunduğunun tespit edilmesi, işaret dili tanıma çalışmalarına da katkı sağlayabilecektir.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF SYMBOLS

| | |
|---|---|
| $S_i$ | The softmax output of $i_{th}$ entry. |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| CAM | Class Activation Mapping |
| CNN | Convolutional Neural Network |
| DCT | Discrete Cosine Transform |
| DNN | Deep Neural Network |
| DWHT | Discrete Walsh Hadamard Transform |
| FCN | Fully Connected Network |
| FN | False Negative |
| FP | False Positive |
| GAN | Generative Adversarial Network |
| IoU | Intersection over Union |
| LBP | Local Binary Pattern |
| N | Negative |
| OALP | Orientation Aware Local Pattern |
| P | Positive |
| PSF | Point Spread Factor |
| RFSV | Response Function on Singular Values |
| SL | Sign Language |
| SLR | Sign Language Recognition |
| SR | Super Resolution |
| SVD | Singular Value Decomposition |
| SVM | Support Vector Machine |
| TN | True Negative |
| TP | True Positive |
| VGG | Visual Geometry Group |
| WSBL | Weakly Supervised Blur Localization |
| WSOL | Weakly Supervised Object Localization |

# 1. INTRODUCTION

Blur is a visual phenomenon affecting media such as photos and videos. It is quite common. It is usually considered to be a defect however, it can also be desired for artistic effect. For instance, "Bokeh" is an effect used to make the object of interest look better by focusing on it and leaving the other parts of the scene blurred. Apart from aesthetic concerns, there are multiple computer vision tasks that may be hindered by blur. Blur may also be useful for some tasks as well. That is why, in an image, knowing if there is blur, if there is blur then knowing where it is and how intense it is may be valuable. Blurry-vs-sharp image classification task is the effort to find which images contain blur. Blur segmentation tries to find blurry pixels in a given image, blur estimation or assessment tries to determine the intensity of blur an image has and deblurring is the effort remove blur from images. Blur is discussed in Chapter 2 in depth.

In this thesis, we concentrate on assessing blur in sign language videos. Sign language (SL) is a visual language, it is the native language of Deaf people. It uses hand gestures, together with facial expressions and upper body gestures. Since the hands can move very fast, blur is frequently encountered in SL videos, this makes sign language recognition (SLR) a more difficult task. Classifying blurry frames, localizing the blur and estimating the amount of blur could help in increasing the success of SLR.

We now present the contributions we make in this thesis. We propose a deep CNN model for sharp / motion blurred classification, we show its performance with extensive experimentation. We see the lack of labeled data as the main reason why deep learning architectures were not utilized for blur level estimation. We propose and show how weak-supervision can be used as remedy for the lack of data. We also demonstrate how weak-supervision can be used with class activation mapping for blur localization.

### 1.1. Organization of This Thesis

The rest of this thesis is organized as follows: In Chapter 2, we discuss blur in depth. In Chapter 3, we discuss motion blurred vs sharp image classification. We introduce our model. In Chapter 4, we discuss motion blur estimation. We describe how we use the model we introduced in Chapter 3 for blur estimation. In Chapter 5, we explain how weak-supervision can be used with CAM for blur localization. In Chapter 6, we present our experiments and the results. Chapter 7 presents our conclusion.

# 2. BLUR

Blur is a fairly common phenomenon. It happens when the camera fails to capture details of the scene being photographed or filmed due to technical mismatch of the camera. For this reason, blur is usually considered to be a defect. Yet, it can sometimes be desired for artistic visual effects.

## 2.1. Types and Causes of Blur

Blur has two main types causes: Out of focus blur and motion blur. Out of focus blur happens when at least some part of the scene being captured falls out of the focus of the camera. The lens of the camera focuses incoming light to the light sensor where it is recorded. If an object is out of the focus of the lens, then the light coming from it to the camera can not be successfully focused and this causes that part of the image to not form properly. The details get distorted and thus blur occurs. Motion blur happens if things being photographed or the camera itself moves during the recording. Exposure time is the time span where the light sensor of the camera captures and records incoming light. During that time, if there is relative motion between the camera and the scene, light from different parts of the scene end up being recorded on top of each other and result in a blurred image. For example, if an object moves from point A to point B during the exposure time, some light from it ends up recorded when it is at point A and some light from it ends up being recorded when it is at point B. This makes the object seem like it was in both points in the final image. But since enough light was not captured to form the object's details and light from the background also got captured from both points A and B in the final image, the final image is motion blurred. Different types of blur have different characteristics, such as texture or homogeneity.

Haze also sometimes considered to be "natural blur". "Gaussian" blur is obtained by simply convolving an image with a gaussian function which smooths the image and results in a blur-like appearance.

Figure 2.1 shows some examples of blur.

Figure 2.1. Blur examples. From left to right, motion blur [1], out of focus blur [2] and haze respectively [3].

Given an image with blur, finding which type of blur it has is called blur classification or blur type identification.

## 2.2. Dealing with Blur

Blur detection and blur segmentation are efforts to find blur in visual media. The main difference is that segmentation assigns blurry/sharp labels to each pixel of the media where detection finds the smallest (normally rectangular) area that will confine the blurred area.

The effort to remove undesired effects of blur is called deblurring. Given a blurry image, deblurring aims to remove the blur to obtain a sharp version of it. It should be noted that in case of defocus blur this makes perfect sense since there is only one latent sharp image for the blurred version of it. However, it is less clear what the sharp version of a motion-blurred image should be. Because moving things in the image are naturally in different places during the capture. Given that, it is not clear where the objects should be in the sharp, deblurred image. Motion can be complex too. The scene may have multiple objects moving and rotating at different velocities. This makes deblurring motion blur an ill-posed problem.

### 2.3. Motivation and Related Work

Intuitively blur can hinder tasks such as classification of images, sign language recognition or human pose estimation on images or videos. Therefore it is valuable to find out if an image or a frame from a video contains blur or not. When we know if an image has blur in it, we can take precautions against the failure of our systems or algorithms which are designed to work on sharp, clear images. For example, deblurring is an effort to mitigate the adverse effects of blur in images. One may try to detect blurry images from a dataset, for instance sign language videos, and mitigate the effects by deblurring those images.

On the other hand, one may use blurry images for classification or pose detection tasks. With the right architecture, blur may not be as significant a problem as occlusion. The intuition about this comes from our observation of the very high success rate of Openpose [6] despite blur, which is a deep learning based solution for human pose estimation. It might be the case that, at least for some problems, the right architecture may be blur resilient or even immune to blur for all practical applications. Yet blur resilience is not the only priority we have. We have got other constraints for the models we use as well. For example, even if a model is virtually immune to blur, if we can not use it on the edge (run these models on small devices with limited processing power such as smartphones) then it may be useless for a lot of applications involving privacy concerns. This is because sometimes due to privacy concerns or very strict time constraints (for real-time applications) we can not send images to a remote server and the only viable option remaining is on-the-edge computing. All these must be taken into account when studying effects of blur.

There are multiple areas of studies involving blur. Blur segmentation / detection, blur classification or blur type identification and deblurring are some of the most common areas.

### 2.3.1. Blur Detection or Segmentation

Detection, in general, involves finding bounding boxes around regions of interest for instance, cars or pedestrians. However, in the blur detection literature, the researchers usually prefer finding pixel-wise blur maps. Which is in fact segmentation. That is why we chose to converse these under a single section.

Early blur detection/segmentation attempts used hand crafted methods such as using texture statistics, gradient based methods, frequency based methods, edge detection and similar. Recently learning-based methods became popular especially deep learning models.

Marziliano et al. [7] proposed a perceptual blur metric that may be used for detecting blur involving parts of images. However, it only relied on edge detection and was tested on only artificially blurred images. The artificial blur was limited to gaussian blur and jpeg compression artifacts.

Kalalembang et al. [8] used a discrete cosine transformation based method for local motion blur detection. Their purpose was to find blurred parts in an image before applying deblurring to those parts instead of applying deblurring to the whole image. They used blocks or patches of fixed sizes to measure the amount of blur on them and experimented with various sizes of blocks. They tried to model motion blur with a point spread factor (PSF):

$$H(x,y) = f^{\alpha} * m = \sum_{K=0}^{K=1} m_K f(x + cosa, y + Ksina) \tag{2.1}$$

where $\alpha$ is the shifting angle and $K$ is the coefficient. Having to set a fixed block size is a disadvantage because what fits in a fixed-sized box changes drastically depending on the resolution of the camera and the distance between the camera and the scene.

Chen et al. [9] offered a method for motion blurred region detection which operates on fixed sized patches. Just like in Kalalembang's work this is a drawback, as mentioned by Chen et al. What fits in a 30 by 30 image crop (they have chosen 30 empirically) may differ drastically from image to image.

Shi et al. [4] carried out a research about discriminative features of blur. In other words, they investigated which features could be used to tell a blurred image region from sharp image region. This work can be thought of as feature engineering or representation learning. Furthermore, they gathered and annotated a dataset of 1000 images and their binary masks for blurred vs sharp pixels. Shi et al. considered three types of features. Image gradient distribution based methods, frequency domain spectra based methods and hand crafted local filters which are convolution based feature extractors like gabor or edge detectors like laplacian. They have also demonstrated the effects of studying the image at different scales (See Figure 2.2). It is true that different scales may affect the perception of blur, yet it is also important to keep in mind that resizing algorithms may mislead a human inspector when scales are shown as such.

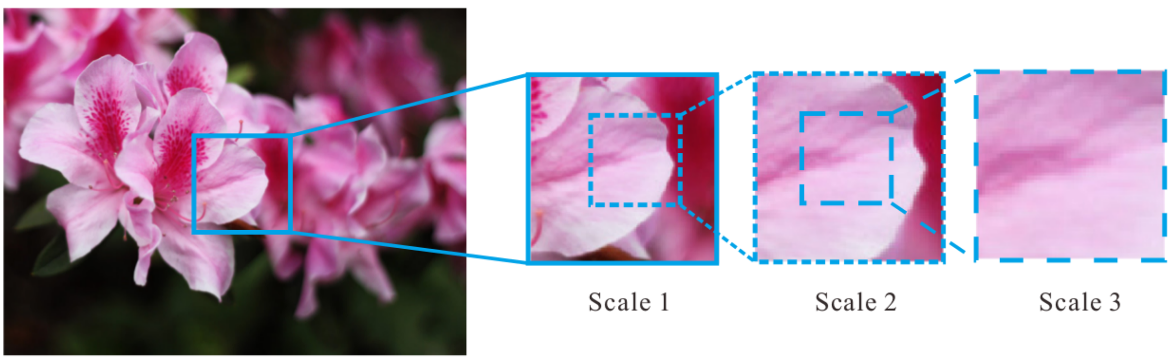


Figure 2.2. Scales used by Shi et al. are illustrated [4] Copyright © 2014, IEEE.

Purohit et al. [10] proposed the first end-to-end learning-based solution for blur detection on single images. The solution utilizes two convolutional neural networks for blur detection. One of them works on patches and the other one works on whole images. Then a third network takes outputs of those networks and outputs segmentation maps. They use the dataset introduced by Shi et al. They also used a synthetically blurred dataset for training-only.

Kim et al. [11] proposed a deep learning based method for detecting defocus and motion blur. The novelty of this work is working on images that have both motion blur and defocus blur. In other words, they detect and classify blur as motion blur or defocus blur. They use CUHK dataset as it has been the only public dataset on the topic as well as their own synthesized dataset. They used an encoder-decoder architecture and performed transfer learning with a pretrained VGG-19 [12]. The encoder-decoder architecture they used has skip connections similar to U-Net [13]. And overall this architecture can even be thought of as a modified U-Net which is an architecture originally used for segmentation. This situation was mentioned by the authors as well. As a novelty, the proposed architecture was trained with multi-scale reconstruction loss functions. The use of multi-scale loss is justified with the "start from coarse then add the details" idea. Yet the results of the ablation study in the paper show that the multi-scale loss only improved the accuracy by 0.008, the F-measure by 0.0095 and mIoU by 0.0152 over the baseline.

In the paper "Defocus Blur Detection via Multi-Stream Bottom-Top-Bottom Fully Convolutional Network", Zhao et al. [14] worked exclusively on defocus blur segmentation. Their method has a multi-scale approach similar to that of Kim et al. The main problematic situations they attack are homogeneous regions which can not be dealt with texture detection or edge detection, low-contrast focal regions and background clutter. They hope that inspecting the image in different scales will reveal if a region is blurred or just has homogeneous color, texture. In the ablation study they have reported that F1 score on the only available dataset [4] they have shown that F-measure did not improve drastically and even dropped when a scale of 0.4 is added along with scales of 1, 0.8 and 0.6. This may imply that using multi-scale alone does not necessarily improve the model in a statistically meaningful manner.

Tang et al. [15] also used a multi-scale architecture. They claim that blur perception is sensitive to scale. They also observe that the boundaries of blur maps are sharp and that cluttered backgrounds make the problem difficult to solve and reduce the performance of the models.

The researchers use two datasets: Shi et al.'s dataset which has been the only publicly available large dataset and another dataset introduced in 2018 by Zhao et al. which has 500 images only.

It is also worth mentioning that they used a pretrained VGG-16 network for extracting features from multiple scales. They used two parallel pipelines, one for shallow features which supposedly holds more spatial information and one for deep features which supposedly holds more semantic information. They train the pipelines in a cross layer manner.

Qian et al. [16] in their paper "Defocus Blur Detection via Salient Region Detection Prior" point out to the fact that the research field terribly lacks publicly available training data. Starting from that point and utilizing the intuition that salient objects in an image are most likely to be in the focus whereas other things in the background will be out of the focus, the authors pretrained their models on salient object detection datasets.

Wang et al. [17] try to solve the blur segmentation problem jointly with the blur kernel estimation problem although only one blurred region from a single blur kernel per image is assumed.

Ali and Mahmood [18] in their work "Analysis of Blur Measure Operators for Single Image Blur Segmentation" focus on blur measure techniques. They too use Shi's dataset. In their experiments, they first applied the blur measure operators to the images. Then they segmented blurry regions from the images. In their work, they have investigated 32 different blur measure operators to find the blur level of each pixel: 11 derivative based operators, 10 statistics based operators, five transform based operators and six other operators.

Wang et al. [19] in their work called "Fast detection and segmentation of partial image blur based on discrete Walsh-Hadamard transform" try to segment blur (motion or defocus) from images fast. Their blurriness metric is based on Walsh-Hadamard transform or more precisely it can be written as,

$$Blur_i = \frac{L_P norm(DWHT(reblur(patch_i)))}{L_P norm(DWHT(patch_i))} \tag{2.2}$$

where DWHT stands for discrete Walsh Hadamard transform and $Blur_i$ means blurriness of pixel i in the center of $patch_i$.

One idea they used is re-blurring. The idea is that artificially blurring a sharp image makes a big difference whereas blurring an already blurred image makes less of a difference. However, if a sharp region has a smooth surface with little to no texture, blurring it may not result in a very significant change either. Or in general, the amount of change for different regions from different images may be very different and as a result, finding an appropriate threshold may be hard for obtaining the binary blur maps. In this paper, the researchers decided on thresholds based on experimentation and statistical analysis.

Shen et al. [20] worked on blurry region extraction which can be thought of as blur segmentation, based on Semantic Segmentation. Their proposed model is an encoder decoder architecture. The encoder is a Resnet [21] and the decoder consists of fully convolutional networks. To be able to train or at least fine tune such a deep architecture the researchers needed lots of data that is not publicly available. Their solution was augmenting Shi et al.'s dataset by cropping parts from the images. Shi et al.'s dataset has 1000 images in total and the researchers obtained more than 7000 image patches from that.

Table 2.1 summarizes the literature of blur detection and segmentation.

Table 2.1. Literature summary of blur detection and segmentation.

| Paper | Keywords | Dataset | Blur type | Date |
|---|---|---|---|---|
| Marziliano et al. [7] | Edge Det. | Unnamed | Gaus., JPEG | 2002 |
| Kalalembang et al. [8] | DCT | Unnamed | Motion | 2009 |
| Chen et al. [9] | Energy Analysis | Unnamed | Motion (synth) | 2010 |
| Shi et al. [4] | Gradients, FT | CUHK | Motion, defocus | 2014 |
| Wang et al. [17] | LBP | CUHK | Motion, defocus | 2017 |
| Purohit et al. [10] | CNN | CUHK | Motion, defocus | 2018 |
| Kim et al. [11] | U-Net | CUHK | Motion, defocus | 2018 |
| Zhao et al. [14] | CNN | CUHK, DUT | Defocus | 2018 |
| Tang et al. [15] | CNN | CUHK, DUT | Defocus | 2019 |
| Wang et al. [19] | Walsh-Hadamard | CUHK | Motion, defocus | 2019 |
| Qian et al. [16] | CNN - Saliency | CUHK, DUT | Defocus | 2020 |
| Shen et al. [20] | CNN, FCN | CUHK | Motion, defocus | 2020 |

### 2.3.2. Blur Classification (Blur Type Identification)

As mentioned earlier, there are different kinds of blur. Some researchers attempted to classify blur in images into categories as motion blur, out of focus blur and natural blur (haze). Some works mentioned under Detection & Segmentation do classification as well. Wang et al. [22] in their work "Automatic Blur Type Classification via Ensemble SVM" consider haze as a third type of blur along with defocus and motion blurs. They use hand-crafted methods to extract features from images then they use an ensemble of support vector machines to predict the type of the blur and in fact, they have "clear" as a category as well. They also introduced a new dataset. Unfortunately images under the motion blur category from their dataset are globally blurred which is not a desired situation in general because motion blur due to moving objects is naturally regional.

Wang et al. [23] in “Blur Image Classification based on Deep Learning” assumed four types of blur: defocus, gaussian, haze and motion. Their proposed system uses convolutional neural networks (CNNs) to identify the type of blur in images. For the data, they have artificial blur for gaussian, motion and defocus blurs. Only for haze they have used real images. They have also tested their model with image patches with real images. They reported a non-trivial more than 4% difference in accuracy between test sets of real and artificially blurred images.

### 2.3.3. Blur Estimation

Blur estimation, measuring or assessment is the effort to quantify the level of blur an image has. Blur level can be an important image quality metric. It can be used for auto focusing in cameras or finding less blurred images in a dataset. Automated and objective blur assessment can be used to replace any task that requires subjective human evaluation, for example, evaluation deblurring systems.

Ali and Mahmood [18] investigated the performance of state of the art blur measure operators in a systematic fashion. They took 32 blur measure operators from the literature and compared their performance in a segmentation based setting. They classified operators into four categories based on their working principle.

- Derivative based operators: Derivatives of images are used to find edges in images. It is assumed that sharp images have more edges in them.
- Statistical based operators: Assumption is various statistics can signal the amount of blur.
- Transform based operators: The problem is studied in frequency domain instead of spatial domain.
- Miscellaneous: Other methods that don’t technically belong to any one of these three categories.

Liu et al. [24] offered an improvement over the local binary pattern (LBP) operator. The improvement aims to incorporate orientation information obtained via gradients. This proposed orientation-aware local operator is used as a preprocessing step. After that, edges are extracted using Toggle operator and this information is used to train a support vector regression model. It is trained using subjective scores as ground truth. It is important to note that the study involved only wholly blurred images troubled by only gaussian blur.

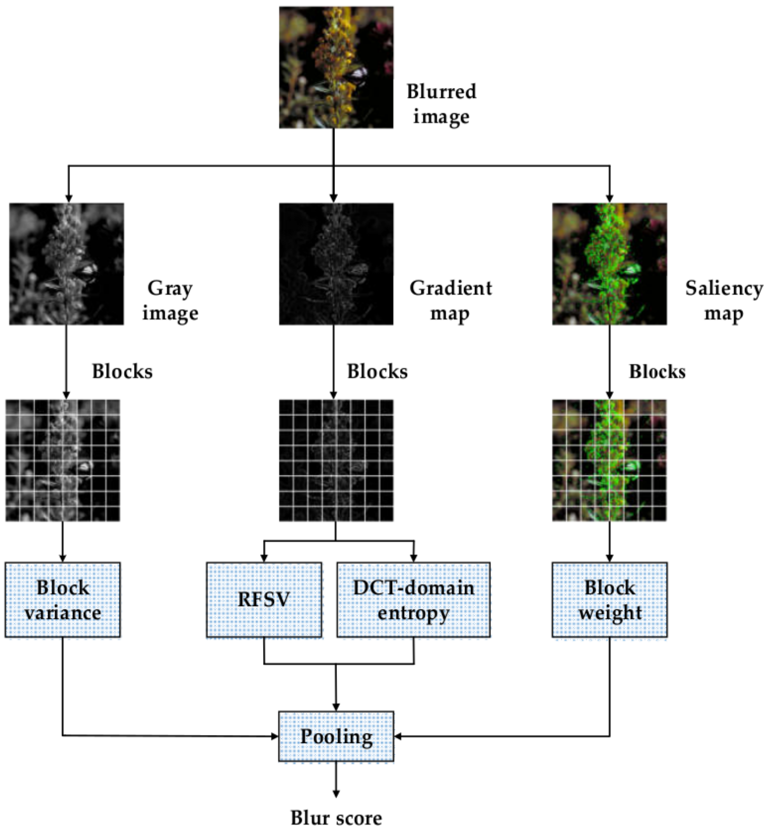


Figure 2.3. Zhang et al.'s [5] solution diagram.

Zhang et al. [5] proposed a solution that uses variance from the grayscale image, gradient map of the image and salience map of the image obtained via scale-invariant feature transform. They first divide the image into blocks, then they calculate block variance from the grayscale image; response function on singular values (RFSV) and discrete cosine transform (DCT) domain entropy from the gradient map and saliency map separately. They finally use these four above mentioned components to calculate a blur score. Variances and DCT domain entropy are utilized for normalizing the RFSV in hopes to decrease of effects of image content whereas block level salience maps are used to push the score towards subjective human evaluation. The whole process is summarized in a diagram, see Figure 2.3.

Yu et al. [25] proposed a convolutional neural network based approach. The images are turned into grayscale and local contrast normalization is applied as preprocessing. After that a relatively shallow neural network is used to output a score. The model is trained with subjective scores as ground truth and wholly blurred images that have only gaussian blur.

Table 2.2 summarizes the literature of blur estimation.

Table 2.2. Literature summary of blur estimation.

| **Paper** | **Keywords** | **Dataset** | **Blur type** | **Date** |
| --- | --- | --- | --- | --- |
| Su et al. | SVD | Unnamed | Motion, defocus | 2011 |
| Vu et al. | Total Var. | LIVE,CSIQ,TID | Mixed | 2012 |
| Yi et al. | LBP | CUHK | Defocus | 2016 |
| Golestaneh et al. | Gradient, DCT | CUHK | Motion, defocus | 2017 |
| Ali and Mahmood | Curation | CUHK | Motion, defocus | 2018 |
| Zhang et al. | Gradient, DCT, entropy | LIVE,CSIQ,TID | Mixed | 2018 |
| Liu et al. | Edge Det., OALP ( LBP) | LIVE,CSIQ,TID | Mixed | 2020 |

### 2.3.4. Blur Kernel Estimation

Blur can be modelled with a convolution operation:

$$y = x * k + n \tag{2.3}$$

where y is the blurry image at hand, x is the latent sharp image, k is the unknown blur kernel and n is the random noise.

The blur kernel is normally unknown which is supposedly the cause of the blur in a given blurry image. Blur kernel estimation is studied for deblurring purposes. If the blur kernel is known or successfully estimated, deconvolution operation can be used to obtain the latent unblurred image. This is called non-blind deblurring. The whole topic of deblurring will be discussed in the next section.

Sun et al. [26] used a coarse-to-fine iterative approach to estimate the blur kernel. They work on "trusted" patches of the image instead of the whole image. These patches are chosen such that they have edges, corners or t-junctions. They reported state of the art performance on uniformly blurred images.

Xu et al. [27] in their work named "Motion Blur Kernel Estimation via Deep Learning" actually do blind image deblurring. They proposed a two part CNN to select edges and sharpen them. They developed this as an improvement over previous methods for the same purpose. The resulting recovered edges are then can be used by already present kernel estimation methods.

Liang et al. [28] in fact work on the super resolution (SR) problem. SR is basically the task of improving the details of an image. One example is when an image is resized-up the details which are not actually in the image are generated by interpolating pixels around. This causes artifacts and artificial patterns. SR tries to improve the quality of this up-scaling process. Another example is deblurring an image to improve its details. Liang et al. also try to remove blur to achieve SR on single images. Their focus is on spatially variant blur. In that situation, different patches of the image have different blur kernels associated with them. That is why estimating a single blur kernel for the whole image will not yield good results in the following deblurring/SR process. They chose a deep learning based solution. Very deep networks have large receptive fields built up by their many layered structure and because of that, they lose spatial information. For this reason, the researchers proposed a network with a smaller (moderate) receptive field. They used a layer they called mutual affine convolutional layer whose main property is that it refrains from increasing perceptive field. They reported good results but also room for improvement. They left using GANs as future work.

Carbajal et al. [29] also do deblurring with deep learning. But instead of building a system that generates deblurred images from blurred ones, they trained a model that finds blur kernels which then can be used to deblur the image. They reported this approach to be more successful than end-to-end deblurring methods.

Pan et al. [30] work on deblurring. Like others, they estimate the blur kernel and use it for deblurring. The unique thing they bring compared to previous studies is that, they face the problem in the frequency domain. Their main contribution is, they try to find the blur kernel by using what they call the phase-only image. This phase-only image is reconstructed from only phase information of the original image, obtained by Fourier transform. They report that this phase-only image carries valuable information about the blur such as its direction and magnitude.

Table 2.3 summarizes of the blur kernel estimation literature.

Table 2.3. Literature summary of blur kernel estimation.

| **Paper** | **Keywords** | **Dataset** | **Blur type** | **Date** |
|---|---|---|---|---|
| Sun et al. [26] | Edges, priors | Synt. | Motion | 2013 |
| Xu et al. [27] | CNN, edges | Sun et al. | Motion | 2017 |
| Pan et al. [30] | Fourier | Mixed | Motion | 2019 |
| Liang et al. [28] | DNN | Synt. | Defocus | 2021 |
| Carbajal [29] | CNN | Köhler, Synt. | Motion | 2021 |

### 2.3.5. Deblurring

Deblurring can be categorized into two parts: blind and non-blind deblurring. Non-blind deblurring depends on a known or at least estimated blur kernel. Of course, the blur kernel is almost never known unless the blur is artificial and estimating the blur kernel is non-trivial. Furthermore, there is no reason why there has to be just one blur kernel causing the blur in an image. For instance, in case of motion blur, if the velocity of the moving object causing blur changes during shooting, the blur kernel will also change. In other words, the blur may vary spatially within even a single image. A more reasonable approach is not relying on a blur kernel. Instead, learning-based methods may be able to "fix" blurred images by using already learned information. An intuitive comparison may be the following, a human artist may be able to understand the content of a blurred image and be able to reconstruct deterred details using his/her common knowledge about the content of the image in a reasonable manner. Similarly, a learning-based model trained on hundreds of cars "knows" what a car is supposed to look like, and given a blurred image of a car it can fill the information gaps which are lost because of the blur. To be able to do the gap-filling operation in a non-arbitrary manner, the model must make use of the general context of the image, the color of the car, its model, in which angle the image is taken, etc. In this regard, deblurring is not so different from image in-painting.

Early deblurring methods aimed to find blur kernels causing the blur and undo or cancel their effects on the images. However, they have not been very effective especially on real images [31]. Later works are utilizing learning-based methods,

Tao et al. [32] attempted to use a coarse to fine enhancement architecture to obtain less blurred images from blurry images. Their network architecture has inspirations from U-Net and multi-scale architectures. They also compare their architecture with those in their paper. The architecture resembles a stack of U-Nets of three different scales. Using a recurrent module in the middle enabled them to use a context vector across scales which supposedly can contain blur kernel information. It may also be more appropriate to say information about the nature of the blur.

Kupyn et al. [31] used a generative adversarial network (GAN) to deblur single images. Their solution also has utilizing multi-scale information in mind. They used a feature pyramid network to get feature maps from different scales which hopefully hold different semantic information. Their proposed architecture also has two discriminators. The first one is called the local discriminator and it operates on randomly cropped patches of images and the second one is called global discriminator, which as the name suggests, operates on entire images.

As Zhang et al. [33] also mention in their paper supervised learning methods need ground truths. For deblurring, this ideally means sharp and blurred versions of the same image. This is of course not found normally. That is why some researchers apply artificial blur to sharp images to get blurred counterparts of sharp images. Zhang et al. in their paper named "Deblurring by Realistic Blurring" emphasize that methods used to blur images which are in turn used to train deblurring networks are not guaranteed to obtain realistic-enough synthetic blur. Based on that, they developed a system that will generate blurred versions of input images. By using these fake-blurred images they trained a model that takes a blurred image and generates a deblurred version of it. This way they have trained two GANs: One for generating blurry versions of sharp images, and one that takes blurry images and generates their sharp versions which is by definition deblurring.

Liu et al. [34] worked on linear motion deblurring. They too took the lack of labels into account and developed a self supervised learning based method. This method requires consecutive blurry frames for training. At test time a single blurred image is enough for deblurring. One particular drawback is this paper only focuses on *linear* motion blur.

Table 2.4 summarizes the literature of deblurring.

Table 2.4. Literature summary of deblurring.

| **Paper** | **Keywords** | **Dataset** | **Blur type** | **Date** |
|---|---|---|---|---|
| Tao et al. [32] | SRN | Gopro | Motion | 2018 |
| Kupyn et al. [31] | GAN | Gopro, DVD, NFS | Motion | 2019 |
| Zhang et al. [35] | GAN | Gopro, RWBI | Motion | 2020 |
| Liu et al. [34] | DNN | Gopro, new | Motion | 2020 |
| Tran et al. [36] | GAN | Gopro, synt. | Motion | 2021 |

# 3. MOTION BLURRED VS SHARP IMAGE CLASSIFIER

Finding images that contain motion blur from a set of images is, in essence, a binary classification task. Deep convolutional neural networks (CNN) have shown to be most effective in image classification tasks [37]. Therefore, we utilize a CNN for blur classification.

### 3.0.1. The Model Architecture

We have experimented with various deep convolutional neural network architectures, which have proven performance on image classification tasks. These architectures include some naive CNNs of a few convolutional layers as well as deeper architectures such as VGG-16, VGG-19 and inception res-net version2 [38] using Keras [39] library.

For learning-based methods, there is a trade-off between bias and variance. For a sufficiently expressive model, it is possible to memorize the dataset instead of learning the underlying distribution. When that happens, as a result, the learned model does not generalize to other datasets and it has poor performance on them. This problem is called overfitting. The opposite problem is called underfitting. It happens when the model is not expressive enough to learn the underlying distribution successfully and inevitably it has below optimum performance. Figure 3.1 illustrates underfitting and overfitting in a binary classification problem. The data points belong to red and blue classes, x and y axes are the only two features of the dataset. Learned decision boundaries of a neural network classifier are visualized as red and blue contours.

In our experiments, we have observed that a version of VGG-16 employing regularization with spatial dropout layers before each max pooling layer has provided a more robust performance while retaining high accuracy (See Figure 3.2). This VGG-16 derived architecture seems to be in the right position that has a better balance of generalization and performance when fine-tuned on our dataset.

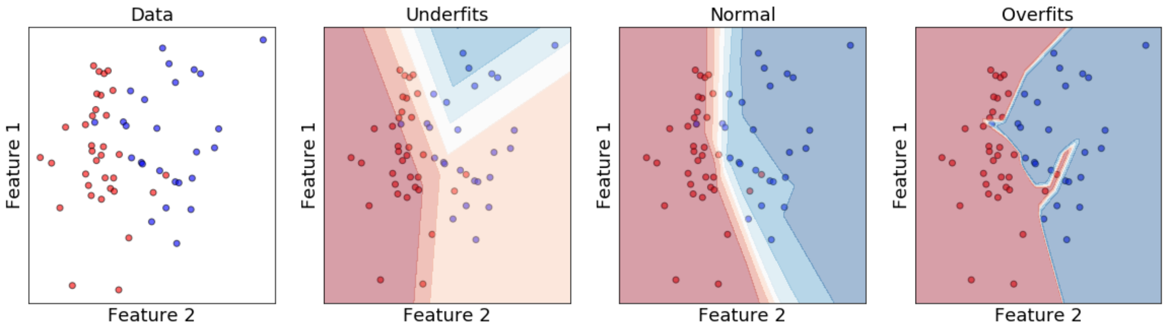


Figure 3.1. The data, underfitting, normal and overfitting respectively.

Dropout method [40] is a simple yet elegant countermeasure against overfitting. Basically, it disables some activations while training, hoping to enforce the network to learn alternative neural pathways to prevent relying on some features excessively. However, classical dropout is rather ineffective in the case of convolutions on images because in a produced feature map, neighboring values are usually correlated. Therefore, dropping out only one of them is not effective. Instead we use spatial dropout [41] which drops out entire feature maps altogether.

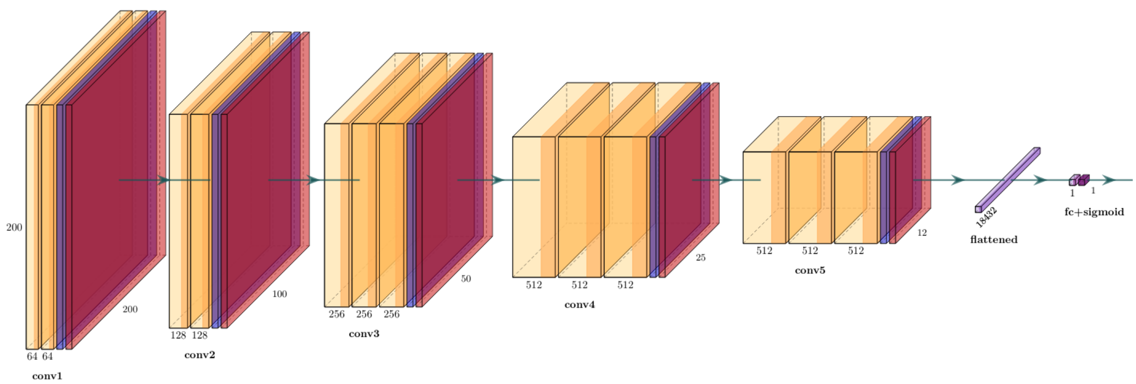


Figure 3.2. The modified VGG-16 architecture.

The architecture we decided on consists of five convolutional blocks and a dense layer of size one on top. The dense layer with one neuron has sigmoid activation function, which couples with binary cross-entropy loss and provides a classifier. Equation (3.1) presents the formula for sigmoid function,

$$S(x) = \frac{1}{1+e^{-x}} \tag{3.1}$$

where $S$ is the sigmoid function and $x$ is a real number. Figure 3.3 is a graph of the sigmoid function.

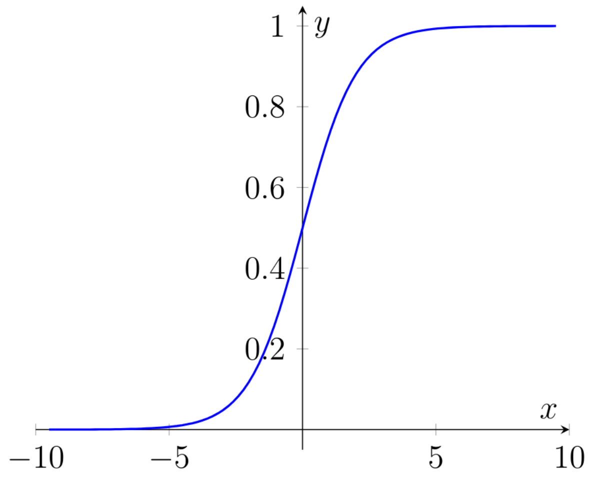


Figure 3.3. The sigmoid function.

Equation (3.2) presents the equation for binary cross-entropy,

$$B(p, y) : (ylog(p) + (1 - y)log(1 - p)) \tag{3.2}$$

where $B$ is the binary cross-entropy function, $y$ is the true label from $\{0, 1\}$ and $p$ is the sigmoid output from the range $(0, 1)$.

Using sigmoid output with binary cross-entropy is a common practice for binary classification. Sigmoid function outputs a single value between 0 and 1.0. Binary accuracy is used with the traditional and common sense threshold of 0.5. Here we have also used soft labels which is a way for dealing with noisy labeled data with the intuition that some images are hard to classify for even human beings. These are images that have very small amounts of motion blur. Unfortunately, in practice, on the validation set using soft labels did not bring considerable performance improvement.

Another benefit of using sigmoid output is that, it can be used as a blurriness (or sharpness depending on the labels) score between 0 and 1. We have qualitatively observed that indeed there is correlation between sigmoid output and human perception of blur-level, although the relation is not necessarily linear. Here, by blur-level we mean how indistinct the blurred part of the image is, not how widespread the blurred region is in the image.

Also, note that the sigmoid function we are using is the logistics function. In a sense, we are doing binary logistic regression and then using 0.5 as a threshold to do classification.

Further labeling data with labels of higher precision, such as five levels of blur, could enable a regression study.

An alternative is using a dense layer of size two instead of one and using softmax activation function with categorical cross-entropy loss function instead of sigmoid activation function with binary cross-entropy loss function. Using an "n"-neurons dense layer with softmax activation function and categorical cross-entropy loss is again a common practice for "n"-class classification. In practice, for n=2, it is equivalent to the sigmoid variant. We have tried both and observed this ourselves as well.

Equation (3.3) shows the equation for softmax function,

$$S(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j))} \tag{3.3}$$

where $S_i$ is softmax output of $ith$ entry in the array of $x$.

We have trained the networks starting from weights pretrained on the ImageNet dataset [42]. In other words, we have used convolutional layers of VGG-16 for transfer learning.

Transfer learning is utilizing weights and parameters learned for a different but usually similar task as a starting point instead of starting from a random configuration. In a sense, the knowledge recorded in the already learned configuration is transferred to another task or model. An example may be using the weights of a human face recognition model to start a model for age estimation.

Of course, the transferred weights are unlikely to be optimal for the new task. That is why the model needs to be fine-tuned. That is, it needs to be trained further to learn the optimum configuration. It makes sense to use small learning rates to avoid disconfiguring already learned weights.

We did transfer learning by first freezing all feature extraction layers and training only the newly added classifier dense layer then, unfreezing all the layers and fine-tuning the whole network together on our data. The reason for using this two step approach is, at first randomly initialized dense layer may cause large gradients and cause some instability in training, i.e. we first make sure that the newly initialized dense layer is caught up with the rest of the network.

# 4. MOTION BLUR ESTIMATION

Blur estimation (also called blur measuring or blur assessment) is the effort to determine how much blur an image has. Unlike blur segmentation or detection, it is not about which part of the image has blur. Blur assessment tries to answer the question: "How deeply troubled is this image by blur?"

There are different approaches to tackle this problem. Different techniques choose different features or use different methods to extract or represent what might be called the blur-features.

Most of the time, blur is simply regarded as a loss of details. Disrupted edges and corners or lack of high-frequency components are taken as features for blur. In some cases, those are actually due to blur, but they do not always signal the existence of blur. Instead, such characteristics, in fact, signal what might be called smoothness.

Ali and Mahmood [18] have curated, studied, and compared many blur measure operators from the literature. Among 32 operators that were studied, we took four of which were reported to have better performance. Also, we have taken a more recent method from Lui et al.'s work [24]. We will present the results of these baselines in the following chapters. However, we will mention common approaches and their shortcomings presently before mentioning our approach.

Gradient based methods look at how fast changes happen in an image. Assuming blur mixes pixels and reduces the rate of change in the image, gradient based methods estimate higher levels of blur for images with low gradients.

Methods operating in the frequency domain, for example, through Fourier transformation, do something very similar. They look at high-frequency components and take the lack of them as a sign of blur.

Both of these approaches would work under some constraints. If other factors affecting the amount of detail in the image, such as the content or the lighting conditions, are very similar in any given two images, then the results these approaches produced for these images would correlate with the amount of blur these images actually have.

Statistical methods analyze different statistical characteristics of blurry and sharp images to find informative features about the blur levels of images. Ratios of some patterns obtained via the LBP operator and variance of pixels are among the utilized statistics. Unfortunately, these statistics also fail to signal the amount of blur successfully in general.

These facts and our observations led us to think that despite what seems to be the general opinion, blur is not a simple characteristic or attribute that we can easily measure or estimate. Under these circumstances, we have accepted that we can not clearly define what blur is or find some mathematical operators that can be used to estimate the amount of blur in any given image. This situation is similar to the case for many fields which have benefited from recent advances in learning based methods, particularly deep learning. For example, although it is so easy for human beings to differentiate cats from dogs, we can not describe how we do it as an algorithm. Deep learning based methods remove the need for an exact algorithm or explicit feature extracting steps because they learn those from the data. Therefore they are a perfect fit for the situation.

Although deep learning has great advantages, it requires a lot of training data. Lack of this data in the "blur" field has been a major problem and we believe this is one of the reasons why traditional methods are still being used in the literature instead of deep learning.

Indeed there is no dataset to the best of our knowledge that can be used to train a blur estimation deep neural network for regression. The problem of blur estimation is not trivial, hence requires a highly expressive model to tackle it, and such models require large amounts of data even if we use transfer learning. To overcome this limitation we, used a dataset that has image-level blurry-or-sharp labels to train a classifier. This classifier has a sigmoid output that can be used as a measure for blur. The details of this network are explained in Chapter 3. Also, the intuition or rationale behind this approach is discussed in Chapter 5.

We have qualitatively verified that the blur level correlates with our model's output. For quantitative results, we used artificial blur. The initial model's results correlated with artificial blur up to a point, but after a level of blur, our estimates started to decrease. We believe this was to be expected because our model is trained on real images with real blur, our dataset does not include extreme blur and after a degree, the blur no longer looks like blur. We tested this idea by fine-tuning the model using only images with extreme blur and have seen that the model also learned extreme blur cases while retaining its performance on normal cases. Figure 4.1 shows the results of the original model and the model that was fine-tuned on extreme blur cases while the details are discussed in Chapter 6.

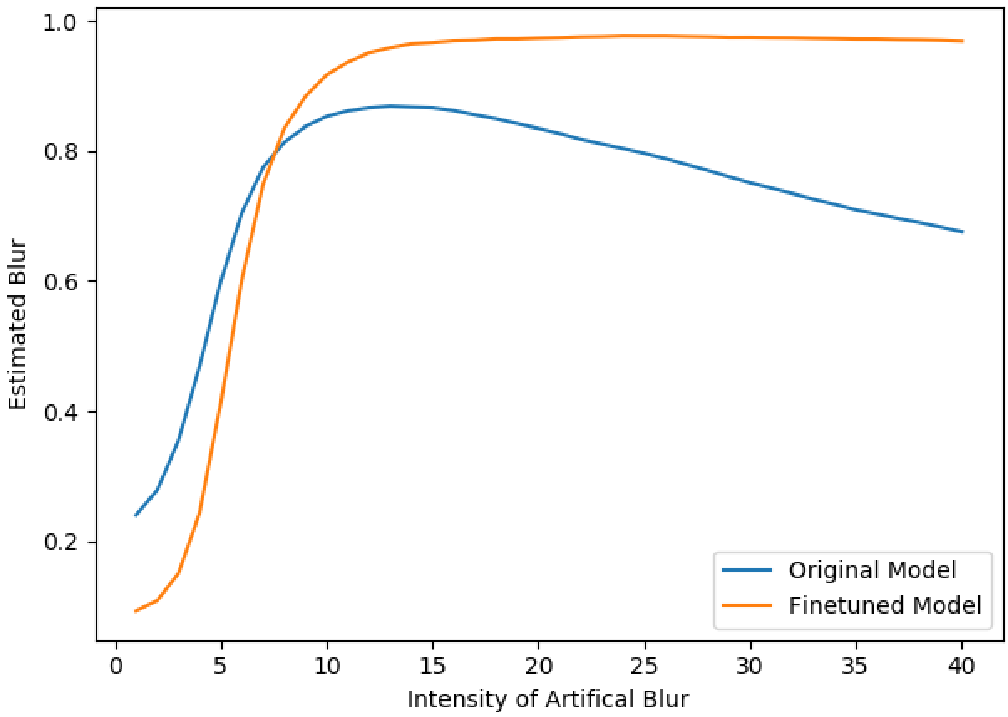


Figure 4.1. Average estimated blur by artificial blur level. The original (in blue) and the fine-tuned (in orange) model outputs.

# 5. WEAKLY SUPERVISED MOTION-BLUR LOCALIZATION

In this chapter, we introduce the concept of weak supervision and illustrate how it can be used as a remedy against the lack of labeled data. We will demonstrate how our model that was introduced in Chapter 3 can be used as a blur localizer with the help of class activation mapping (CAM). We will also provide intuition about how a classifier can be used as a regressor as we did in Chapter 4.

Supervised learning refers to using ground truth annotations for training a machine learning model. An example is using ground-truth binary masks for image segmentation. Whereas weak supervision uses imprecisely annotated data for supervised learning. An example of that is using image-level labels instead of providing binary masks for segmentation or localization tasks. This approach is favorable because annotating binary masks is more cumbersome and costly compared to annotating image-level labels.

In our setting, we did not have binary masks for blurry and sharp parts of images for training a blur segmentation/localization model directly. Instead, we used the model we trained on image-level labels for containing blur or not and used CAM to obtain the localization maps.

CAM was introduced in 2016 and was used for weakly supervised object localization (WSOL) by Zhou et al. [43]. It is used to find which parts of an input-image triggers activation of the neurons of the neural network. Deep CNNs effectively work as detectors. Neurons learn to activate when they encounter patterns informative about the task they are trained for. As a natural consequence, even if the network is trained for classification or regression, what the network implicitly does is detection. Figure 5.1 illustrates this.

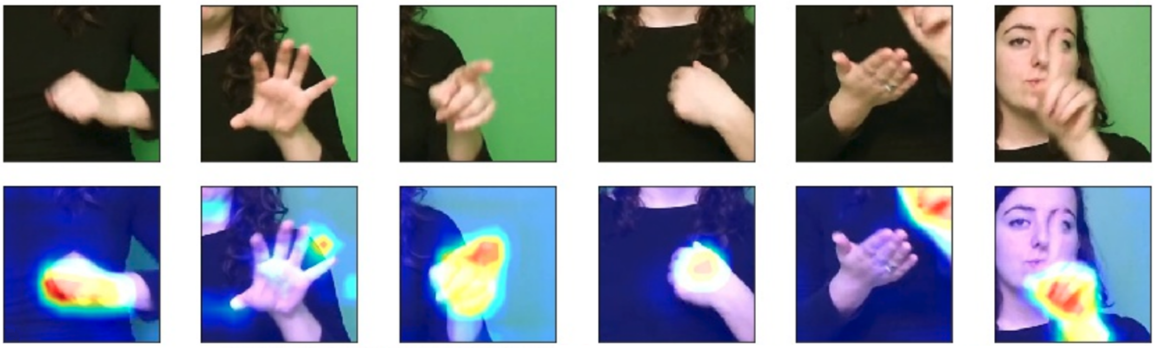

Figure 5.1. Weakly supervised blur localization example.

We did weakly supervised blur localization to illustrate weak supervision is possible and potentially useful in our context. We did it also to see what the model was focusing on as a way of verification that the model has learned to "see" blur. Yet we tested our model to see how it performed according to WSOL metrics as well. To test our model for weakly supervised blur localization we have annotated 60 blurry images with binary masks. Our method achieved 29% IoU score for blur category and has 90% IoU score. Which is low compared to state-of-the-art WSOL results [44]. However, objects normally hold a compact space in an image, on the other hand blur, at least motion-blur in our case, has more intricate and sometimes discontinuous shapes. Where activation maps can not really produce such shaped heatmaps. This may be a reason why our results were not as good compared to the results found in the WSOL literature. It can also be mentioned that segmenting blur is harder for even human beings, whereas detecting an object from the background is trivial.

To make a fair comparison, we also trained a U-Net with 30 images as the researchers did in the original paper of the architecture. We trained it with 30 of our binary-mask-annotated images and tested the model with the remaining 30 images. On our dataset, U-Net performed poorly as well.

These experiments have shown how weak supervision can be useful for localization. And just like localization, weak supervision can be used for regression tasks as well. The deep CNN architecture will work as a detector implicitly, and it is possible that the more blurred the image is, the more neurons will activate and the resulting output will be higher as desired.

It is best to note that weak supervision is less likely to be favorable against normal supervised learning if sufficient amount of precisely annotated data is available. As discussed by Choe et al. [45] weak supervision is ill-posed. Since the network is not guided precisely where to look, it is easier for it to learn undesired, accidental relations and overfit as a result. The details and proof of this are discussed in Choe et al.'s work as well. However, there is reason to believe that our data are particularly suitable for weak supervision. The reason is, the ideal case for weak supervision is when the only differences between classes are the valid and informative features. In our dataset, the only real difference between the two classes is involving blur or not. There are no other significant differences that could mislead the model. For example, for any particular user, if we had only blurry images in our dataset, the model could have associated that user with the "blurry" label and learned to search for that user to label images as "blurry" as an alternative for actually looking for blur in the images.

In Chapter 6, we show that the experimental results align with the intuition we have discussed above.

# 6. EXPERIMENTS AND RESULTS

In this chapter we present our experiments and the results.

## 6.1. Datasets

### 6.1.1. HospiSign

BosphorusSign dataset consists of Turkish Sign Language videos [46]. In each video, a person whom we sometimes refer to as a “user” performs Turkish sign language signals and motions. HospiSign dataset is a subset of BosphorusSign dataset where the videos are about common situations encountered in hospitals.

We have used HospiSign dataset in two different ways. In the first one, we have taken crops from 41 different sign language videos of HospiSign dataset. In those videos, there are six different native Turkish sign language users. They are performing common communication which can happen in a medical facility such as a hospital. These crops are binary-labeled as sharp or blur-involving [47]. Table 6.1 summarizes the data. See Figure 6.1 for some examples. In the second way, for the weakly supervised blur localization (WSBL) task, we have annotated 60 blurry images from HospiSign. All 60 frames are blurry images from user four, which is the test user.

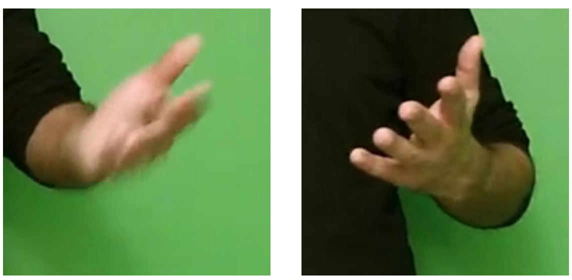

Figure 6.1. Example crops from HospiSign. A blurry example is on the left, a sharp example is on the right.

Table 6.1. HospiSign dataset.

| Data / User | **# of Blurry Samples** | **# of Sharp Samples** |
|---|---|---|
| **User 2** | 6002 | 5563 |
| **User 3** | 5310 | 4668 |
| **User 4** | 5097 | 4448 |
| **User 5** | 5380 | 6292 |
| **User 6** | 3000 | 2089 |
| **User 7** | 2943 | 2406 |
| **Total** | 27.7K | 25.5K |

Figure 6.2 shows some examples from the data.

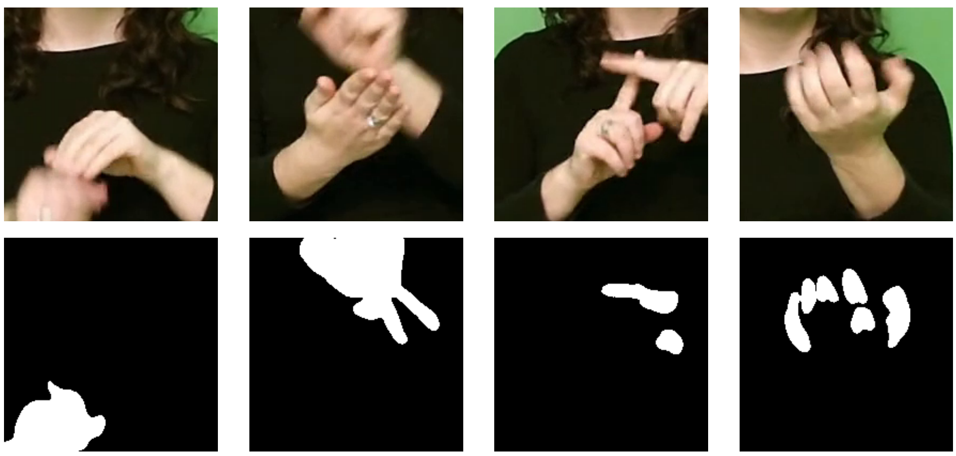

Figure 6.2. Blur segmentation examples. The data and the corresponding ground truths.

### 6.1.2. Phoenix

RWTH-PHOENIX Weather dataset contains weather forecast videos from the German tv station named PHOENIX hence the name [48]. We used frames from this dataset as an additional test set for the blurry-vs-sharp image classification task. There are many videos in the dataset but for our purposes, we selected four videos arbitrarily and labeled some arbitrarily chosen frames from them as blurry or sharp. We annotated a total of 612 frames. The video quality is rather poor in this dataset, that is why the number of frames that are not very troubled by blur is low. We have annotated 495 frames as blurry whereas only 117 frames were annotated as sharp. See Figure 6.3 for examples.

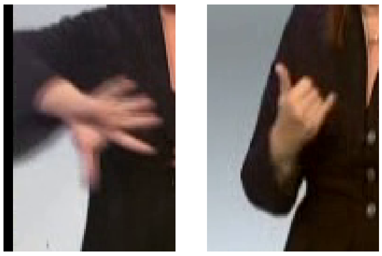

Figure 6.3. Example crops from Phoenix. On the left: blurred, on the right: sharp.

### 6.1.3. Meine DGS

Meine DGS dataset consists of German sign language videos on a variety of topics and formats, such as free conversation or story re-telling [49]. Just like Phoenix dataset, we used frames from Meine DGS as an additional test set for the blurry-vs-sharp image classification task. We have arbitrarily chosen 662 videos from the dataset, and from these videos, we have extracted every 1000$^{\mathrm{th}}$ frame to crop hands. In total, we have labeled 2049 hand crops from the dataset, 1064 of which are sharp and the remaining 985 are blurry. See Figure 6.4 for some examples.

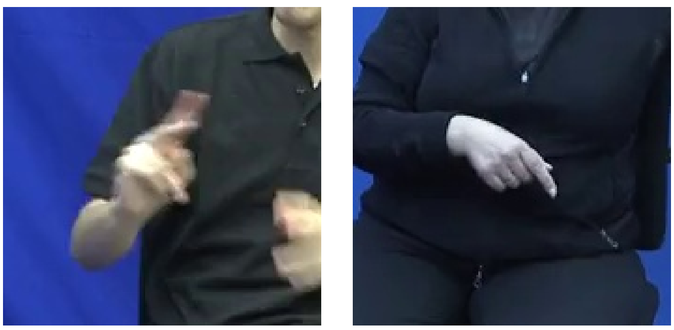

Figure 6.4. Example crops from Meine DGS. On the left: blurred, on the right: sharp.

### 6.1.4. PKU-Multi Modality Dataset (PKU-MMD)

PKU-MMD dataset is a human action detection dataset. In the dataset, 66 users (subjects) perform actions from 51 different categories [50]. We have taken crops from arbitrarily chosen frames from this dataset to synthesize a test set for the blur assessment task. We have labeled 142 frames as sharp images. Then applied different levels of blur to these sharp frames to form a dataset that can be used to measure the performance of our model.

We have applied artificial motion blur with different kernels to simulate different intensities of motion blur. We used 39 different blur levels which makes 5538 images.

## 6.2. Metrics

There are six terms used to explain classification performance metrics. We will introduce these terms now and use them while introducing the metrics in the coming subsections.

The six terms are related to the success of a prediction made by the model for a single data item. In the following definitions, positive refers to the class of interest whereas negative refers to all the other classes. In binary classification case, there is only one negative class. The terms are as follows:

- Positive (P): Data items from the class of interest.
- Negative (N): Data items from the classes **other than** the class of interest. In a sense, these items labeled as negative by the model are rejected, they are deemed "not positive" by the model.
- True-Positive (TP): The data item is correctly classified as positive, i.e. belonging to the class of interest.
- True-Negative (TN): The data item is correctly classified as negative, i.e. belonging to a class other than the class of interest.

- False-Positive (FP): The data item is incorrectly classified as positive.
- False-Negative (FN): The data item is incorrectly classified as negative.

A good real-life example would be a test for a virus. Some laboratory tests are performed on samples (data points) from people to predict if those individuals are infected by the virus. Here, the laboratory test is the "classifier model" and the class sought is the "Infected" class, that is why it is called positive. Table 6.2 shows the possible cases.

Table 6.2. Classification output possible cases.

| Reality / Predicted | **Infected** | **Not Infected** |
|---|---|---|
| **Infected** | TP | FP |
| **Not Infected** | FN | TN |

### 6.2.1. Binary Accuracy

Accuracy is possibly the best known classification metric. It is simply the percentage of correctly classified data points. Binary accuracy is simply a special case for binary classification. The formula for binary classification can be found at Equation (6.1)

$$A = \frac{TP + TN}{P + N} \tag{6.1}$$

where A stands for accuracy.

### 6.2.2. Precision

The model or test predicts "positive" for some data items. Precision shows how careful the system is while giving a data item a "positive" label. Precision drops as the number of false positives increases. The formula for precision is in Equation (6.2)

$$Pr = \frac{TP}{TP + FP} \tag{6.2}$$

which is equal to

$$Pr = \frac{TP}{P} \tag{6.3}$$

where Pr stands for precision.

### 6.2.3. Recall

The model or test tries to find real positive cases while rejecting instances of other classes. Recall shows how careful the system is while rejecting data items i.e. labeling them as negative. Recall decreases as the number of false negatives increases. The formula for recall is in Equation (6.4)

$$R = \frac{TP}{TP + FN} \tag{6.4}$$

where R stands for recall.

### 6.2.4. F1 Score

Accuracy gives an overall estimate for the performance of a model. Unfortunately, it is a healthy metric only when the data are balanced. On the other hand, precision and recall look at different aspects of the performance of the model and neither gives an overall estimate for the quality and performance of the model. That is why another score derived from precision and recall is needed to provide a point estimate for the performance of the model. F1 score is simply the harmonic mean of the precision and recall which can be used for that specific purpose. The formula for it is in Equation (6.5)

$$F_1 = 2\frac{Pr \times R}{Pr + R} \tag{6.5}$$

where $F_1$ stands for F1 score.

### 6.2.5. Intersection over Union

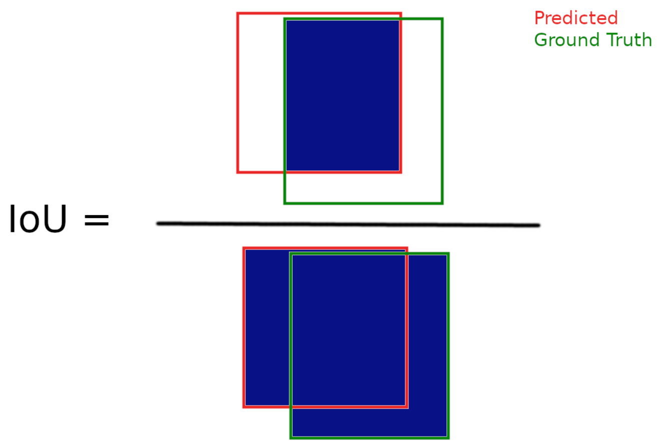


Figure 6.5. Intersection over union. The red rectangle is the predicted area, the green rectangle is the ground truth.

Intersection over union is a metric used to evaluate the performance of detection or localization methods. It is defined as the ratio of intersection of the predicted area and the ground truth to the union of them. It is best explained with a visual, Figure 6.5 illustrates the metric.

### 6.2.6. Bhattacharyya Coefficient

Bhattacharyya coefficient is a measure of similarity between two distributions. It looks at how much two distributions overlap. If two distributions are identical, then they overlap 100%. If they are completely separable then, the Bhattacharyya Coefficient for the pair is zero. Figure 6.6 is an example. There are two distributions, namely blurry and sharp. The overlap is 9.7%.

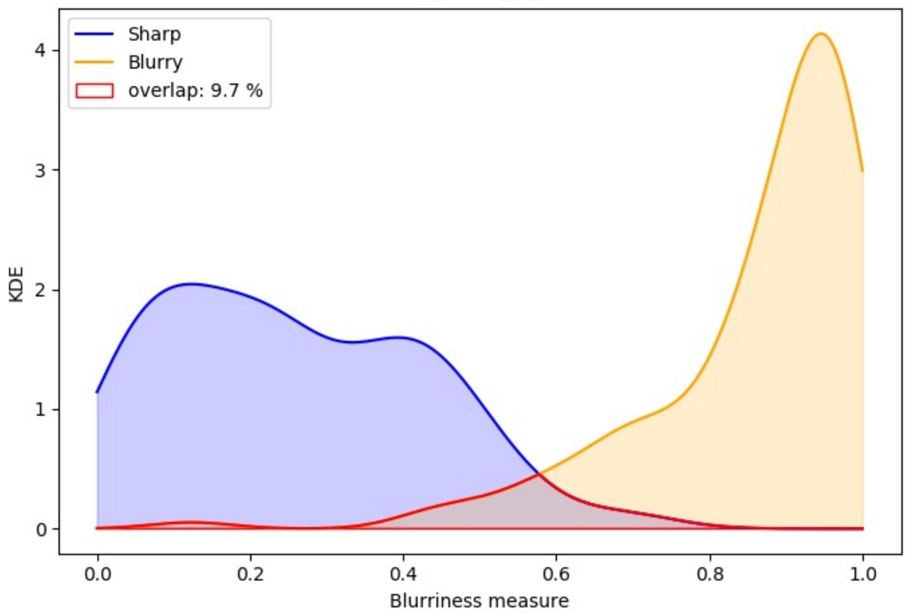


Figure 6.6. Bhattacharyya Coefficient example. The $y$ axis is the kernel density estimate (probability distribution), the $x$ axis is the estimated blur level.

### 6.2.7. Spearman's Rank Order Coefficient

Spearman's Rank Order Coefficient (SRCC) is a statistical metric used to determine how monotonic the relation between given two ranked variables is. It is used in the blur assessment literature [5, 24, 25]. When images are sorted from sharp to blurry, the images are then sorted by the blur assessment models as well then the SRCC is calculated between the models' ordering and the human annotators'. SRCC takes a value between minus one and one. One implies that the given orderings are very well aligned, minus one implies that the orderings are reverse and zero implies there is no link observed between the rankings.

## 6.3. Results

### 6.3.1. Motion Blurred vs Sharp Image Classification

Since deep learning models have acquired state-of-the-art results in image classification tasks recently, we have focused our experiments on deep learning models as well.

We have experimented with unnamed CNNs of a few layers as well as deeper architectures such as VGG-16, VGG-19, and Inception-Resnet-Version-2. We have found a modified version of VGG-16 to be robust and expressive enough. This architecture was explained in Chapter 3.

In our experiments, we have used well-known methodologies for training machine learning models. We have split our dataset into training, validation and test sets. We set apart the test set from the beginning and used the training set to train the model and used the validation set to check if the model actually learned and did not memorize noise. In this process, keeping a fixed validation set has some risks. It is possible to find such hyper-parameters that the model achieves good results only with the fixed training-validation set pair. The ideal set of hyperparameters should be resilient to reasonable changes in the training and validation data. A technique employed to check for that is called n-folds cross-validation. The technique divides the training data into n folds and trains the model n times, each time using n-1 folds for training and remaining one fold for validation. This eliminates the risk of choosing an unfortunate training-validation set pair. It also provides a better estimate and a standard deviation for the validation accuracy. We did not use n-folds cross-validation for hyperparameter search because the process requires us to train the model n times for each configuration of the hyperparameters and we had limited resources. But when we were confident that our model should perform well, before moving on to actual tests, we divided our data (except the test set) into five folds and just like five folds cross-validation we trained the model five times using four folds for training and one remaining fold for validation.

We did not tune any hyperparameters but by this way we made sure that our results were not dependent on the choice of the validation set before moving on the real tests. Once we have seen that the validation accuracy was not dependent on the choice of the validation fold, we moved on to the tests using the model we had trained with the original training-validation sets.

When training deep neural networks, validating and testing the model as objectively as possible is critical because deep learning models can easily memorize the training data and overfit. We have trained our model on the frames from HospiSign dataset. The data are listed in Table 6.1. Since our data consist of frames from videos of six different users, to prevent bias we needed to make sure data of any user did not split into different sets. We kept aside the data of user four as the test set and used the data from user six as the validation set during hyperparameter tuning.

Apart from the test set we kept aside from HospiSign dataset, we have used additional test sets from Meine DGS and Phonenix datasets.

Table 6.3 shows the accuracy results. We see that accuracy on training, validation and test sets from HospiSign are quite close to each other. This is a good sign for the generalization capacity of the model. The model initially had around 77% accuracy on Meine DGS dataset. Since our data size was not too small, we fine-tuned the model on Meine DGS dataset and achieved the same level of accuracy as we had achieved on the original test set. Instead of splitting the 2k images into three sets as training, validation and test. We divided it into four folds and did four-folds cross-validation as a test (we did not search for hyperparameters). The resulting average accuracy is 87% and the standard deviation is 0.028. Since the data from Meine DGS also consist of frames from videos, four videos precisely, we separated the data accordingly and made sure that frames from any video did not get into both fine-tuning and testing folds. All in all, we have seen that our model can perform well on Meine DGS too.

Frames from Phoenix dataset are small compared to what the model needs as input. When zero-padding or resizing is applied to the frames to match the input size of the model, the accuracy dropped about 7%. We did not perform fine-tuning because we did not have much data and the results were not too bad. Accuracy was actually on par with the original test set when edge padding was applied.

Table 6.3. Accuracy of Blur/Sharp classification.

| Data \ Variable | **# of Samples** | **Accuracy** |
|---|---|---|
| **Training (HospiSign)** | 38k | 87% |
| **Validation (HospiSign)** | 5k | 85% |
| **Test (HospiSign)** | 10k | 87% |
| **Meine DGS** | 2k | 87% |
| **Phoenix Zero Padded** | 612 | 80% |
| **Phoenix Resized** | 612 | 81% |
| **Phoenix Edge Padded** | 612 | 88% |

HospiSign and Meine DGS are balanced between classes but Phoenix dataset is skewed. Since accuracy is not a good estimate of performance when the data are skewed, we used other metrics such as F1 score to monitor the performance of the model on Phoenix dataset. Table 6.4 shows results for HospiSign as well.

Table 6.4. HospiSign dataset results.

| Class \ Metric | **Precision** | **Recall** | **F1 Score** | **Data Size** |
|---|---|---|---|---|
| **Blur** | 93% | 81% | 87% | 5097 |
| **Sharp** | 81% | 93% | 87% | 4448 |
| **Weighted Average** | 88% | 87% | 87% | 9545 |

The frames of Phoenix dataset are smaller than the size that the model requires as input, that is why they must be enlarged. Since different enlarging methods induce different effects on the data, we experiment with more than one of them.

Table 6.5 shows the results when zero padding is used. Table 6.6 shows the results when rescaling with bicubic interpolation is used. Table 6.5 shows the results when edge padding is used. Edge padding simply repeats the pixels in the borders until the desired size.

Table 6.5. Phoenix dataset results with zero padding.

| Metric / Class | **Precision** | **Recall** | **F1 Score** | **Data Size** |
|---|---|---|---|---|
| **Blur** | 99% | 77% | 86% | 495 |
| **Sharp** | 49% | 96% | 65% | 117 |
| **Weighted Average** | 89% | 80% | 82% | 612 |

Table 6.6. Phoenix dataset results with rescaling.

| Metric / Class | **Precision** | **Recall** | **F1 Score** | **Data Size** |
|---|---|---|---|---|
| **Blur** | 81% | 100% | 90% | 495 |
| **Sharp** | 100% | 1% | 2% | 117 |
| **Weighted Average** | 85% | 81% | 73% | 612 |

Table 6.7. Phoenix dataset results with edge padding.

| Metric / Class | **Precision** | **Recall** | **F1 Score** | **Data Size** |
|---|---|---|---|---|
| **Blur** | 89% | 97% | 93% | 495 |
| **Sharp** | 81% | 51% | 63% | 117 |
| **Weighted Average** | 88% | 88% | 87% | 612 |

We have experimented with other methods for enlarging the images but we only report the most logical ones to use.

We see that zero padding reduces precision for sharp class and drops recall for blur class. We believe this is because zero padding generates false edges that confuse the model. Blurry images are labeled as sharp by mistake. This reduces precision for sharp by increasing the number of false positives for sharp and reduces recall for blur by decreasing true positives for blur.

Rescaling with bicubic interpolation makes the images appear blurry. This maximizes recall for blur at the expense of recall for sharp. Basically, almost every image appears blurry to the model.

Finally, edge padding which solves the problem of false edges caused by zero padding gives good results except for the decline in the recall for sharp. We believe repeated pixels on the borders cause a smoothness effect which is mistaken with blur by the model in some cases.

We believe more data from Phoenix dataset which would enable fine-tuning or a specialized padding schema could improve the results.

In our early experiments, we have tried using soft labels because some of our data were hard to label, even for human beings. We thought using soft labels could improve accuracy, but in our experiments, we did not observe any significant improvements, and soft labels even seemed to cause some performance drops sometimes. That is why we saw no reason to continue utilizing soft labels and abandoned the idea early on.

### 6.3.2. Weakly Supervised Motion-Blur Localization (WSBL)

We have employed CAM for WSBL. CAM shows which parts of the input image activate the neurons for each class. In our case, the "class" was blur. WSBL provided two benefits to us. The first of them is, it proved that weak supervision can yield results in our context and the second one is, it provided a way to debug our model to see what the model was focusing on when making the decisions.

Visual inspection showed that the model could find the blurry areas in the images. We have also observed that some of the blurry areas could be missed when other blurry parts are enough for the model to classify the image as blurry. Figure 5.1 shows some examples. This is a shortcoming of weak supervision.

Apart from visual/qualitative inspection, we have made some quantitative experiments as well. To be able to do that, we have annotated 60 images from the HospiSign test set. Our method achieved 29% IoU score for blur category and 90% IoU score overall. This performance is low compared to what is reported in the WSOL literature.

It is worth mentioning that object localization is trivial to human beings in most cases and disagreements between human beings for the ground truths are rare. However, localizing or segmenting blur is more difficult and disagreements happen fairly often while annotating the ground truths. We believe this is partially the reason why the scores are low compared to the object localization results found in the literature.

Aiming to have a baseline on our data for WSBL, we also trained a U-Net with 30 of our images and tested it on the remaining 30 images, the performance was too low to report. This is again, we believe, partially due to the same reasons we mentioned above.

### 6.3.3. Motion Blur Estimation

We will first describe our proposed model and other metrics from the literature. Then, we will present the results.

For motion blur estimation we used the model we described in Chapter 3. We trained it in a weakly supervised manner. We trained it as a binary classification model using binary cross entropy loss. For blur estimation, we use the sigmoid output of the model. The sigmoid output is a number between 0 and 1.0. In theory, this output can be interpreted as a measure of the level of blur intensity. Our experiments show that this is true in practice as well.

We have compared the performance of our proposed architecture for the given task against other methods using blur metrics proposed in the literature before. Ali and Mahmood compiled blur measuring metrics and compared them. We have taken four of those which were reported to have better performance as a baseline. These four metrics are as follows:

The first one is a singular value decomposition (SVD) based metric. Su et al. [51] utilized a SVD based blur measure, their proposed method works in a window around a pixel. The SVD is computed for that window and the ratio of the largest $k$ singular values to all of the singular values is used as the blur-measure for the central pixel. The intuition for this method is, the smaller singular values are supposed to represent fine details and the rest of them are supposed to represent the general features. The formula is in Equation (6.6)

$$M_{SVD} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{6.6}$$

where $n$ is the total number of singular values.

The second one was proposed by Vu et al. [52]. The researchers proposed a total variation based method for measuring blur. The method looks at the change in the intensity levels of pixels in a small window. The image is converted to grayscale, the pixel values are normalized to a scale of 0 to 1. Total variation of even smaller patches are computed within the window and the maximum of them is picked. The taken value is normalized to the scale of 0 to 1 by division by the patch size. For example, if the patch is two by two, then the largest possible total variation for the patch is four and we need to scale it down by dividing it by four.

The third metric approaches the problem in the frequency domain. Golestaneh and Karam [53] utilizes discrete cosine transformation (DCT). Around each pixel, the researchers take patches of different sizes, apply DCT to get the frequency components for all patches and aggregate the components together. They sort these components then divide them into groups. They normalize each group between 0 and 1. The blur measure they use is the maximum of the normalized coefficients.

The fourth and final metric we have taken from Ali and Mahmood's curation was proposed by Yi and Eramina [54] and it is an LBP based method. They have observed that some patterns are more common in blurred images than sharp images. These common ones are the ones with values from six to nine. They used this feature as a measure for blur. First, they calculate rotation invariant LBP for image patches then, they calculate the percentage of the number of patterns with values from six to nine within all the patterns. In other words, they use the ratio of patterns more commonly found in the blurry images to all patterns (the number of pixels) as the blur measure.

Liu et al. [24] also proposed an LBP based method. They use a modified LBP operator to get what they call orientation-aware local patterns. Then they use Toggle operator to get the edge information from the image and use that information as weights for the local patterns they had calculated. They use these as features for a support vector regression model which they train using subjective blur scores as ground truth.

Some of the methods we have mentioned calculate local blur scores. They calculate a score per pixel or neighborhood. To get image-level blur measures, we used sliding window averages on pixel/neighborhood-level blur-measures and took the maximum of them as the blur level of the image. Since we can control the size of the blurred area in the artificial test set, we matched the size of the window to the size of the blurred area to make sure these methods are not badly affected by this operation.

All of these methods technically give outputs between 0 and 1. But in practice their ranges of output are more limited and some of the methods are not directly comparable with each other for that reason.

We compare the performances of these methods using artificial blur and Bhattacharyya Coefficient as the metric.

We have taken sharp images from PKU-MMD dataset which we have not used until this point and applied 39 different levels of blur to form 39 artificially-blurred datasets. To get an estimate of a methods performance, we do the following steps:

We apply the method to each blurred dataset to get blur scores. We also apply the method to the original sharp dataset. We end up with 39 blur score distributions from blurred datasets and one blur score distribution from the sharp dataset. We take each of the 39 distributions one by one and calculate the Bhattacharyya Coefficient between them and the sharp distribution which gives us 39 Bhattacharyya Coefficients for the method. Bhattacharyya coefficient shows how much the given two distributions overlap. If the method used to form these blur distributions works very effectively, then the separability of the distributions should increase as the blur level increases. Hence the Bhattacharyya Coefficient should decrease.

We plot how the Bhattacharyya Coefficients change as the artificial blur level increases for each method to see how they perform. See Figure 6.7. In the figure, the $y$ axis is the Bhattacharyya Coefficients (intersection between distributions), and the $x$ axis is the blur level (the numbers on the $x$ are actually the kernel size used for the artificial blur, the axis starts from two since a kernel of size one would be pointless). When there is barely any blur, the distributions mostly overlap naturally. As the intensity of the blur increases, we observe that all the methods start to be able to separate the distributions, some perform better than others. Our initial results (plotted brown, tagged "Ours") had some problems for extreme levels of blur. We believe this was because our training data contained only natural blur and it did not contain any cases with extreme blur. As the blur level increases, blur does not look like natural blur after a point. To verify that, we fine-tuned our model with extreme blur. We did it in the following way: We took sharp training-set-images from HospiSign dataset and applied extreme blur to them. Although we could do regression, to not to stray from weak supervision we fine-tuned the model using a classification setup as before. The results confirmed our beliefs, which can be seen in the graph.

We also used SRCC to test the performance of our model. We did this in two ways. First, we used the datasets we generated with artificial blur. We have seen that all of the methods' outputs have very high (above 0.99) SRCC values with respect to the level of artificial blur. This shows that all of the methods work and they can tell how much blur an image has if the underlying content is the same. Of course in reality this is not the case. That is why we used the SRCC metric in a second way. We ranked 30 images with real blur from our test set and compared how they align with the methods' rankings. The results signals that our method out performs the others. Table 6.8 shows the results.

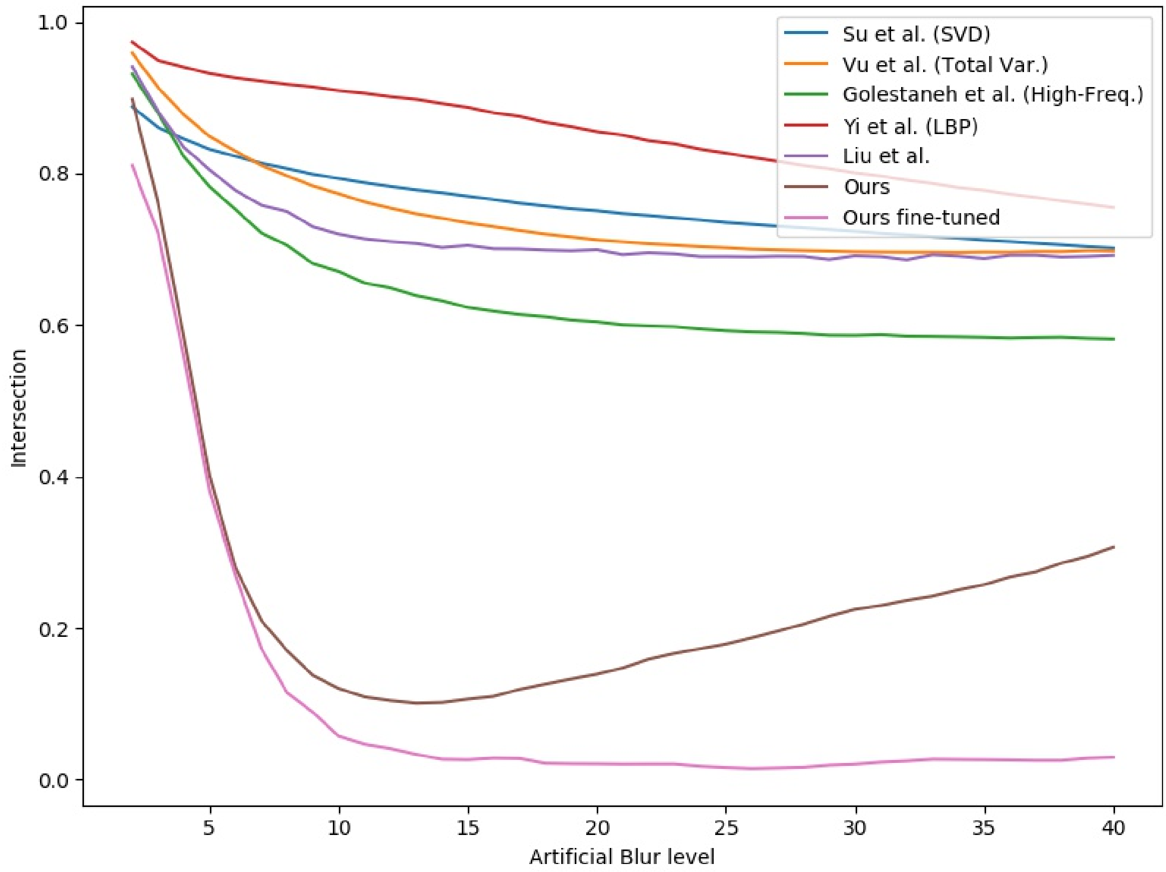


Figure 6.7. The comparison of models.

We investigated on which frames our model erred the most. We have seen that our model seems to do poorly if blurry part of the image is smaller compared to the those of the others even if the blur is intense. Figure 6.8 shows some examples.

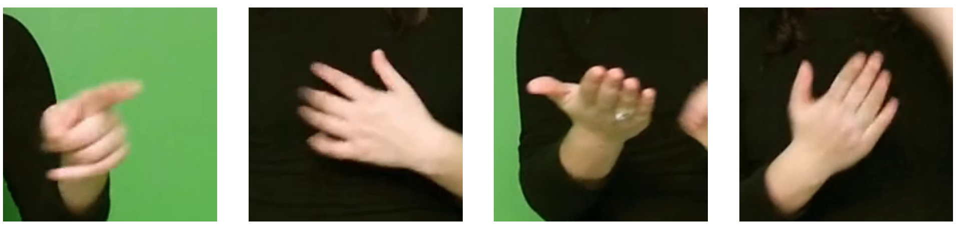

Figure 6.8. Failure cases.

Figure 6.9. Failure cases of others.

Table 6.8. Spearman's rank order coefficient results.

| Method | SRCC |
|---|---|
| **Su et al. (SVD)** | 0.060 |
| **Vu et al. (Total Var.)** | 0.031 |
| **Yi et al. (LBP)** | 0.114 |
| **Golestaneh et al. (DCT)** | 0.232 |
| **Liu et al.** | 0.323 |
| **Ours** | 0.373 |

Mistakes of other methods vary but two examples seem to be common. See Figure 6.9. The first example from left is also a common problem with our model. The example on the right is successfully deemed highly blurry by our model however other methods under estimate the intensity of blur on that image. They too agree that this image has blur, but their estimated blur level is low compared to human judgement.

# 7. CONCLUSION

In this thesis we discussed blur in depth. We presented the literature around blur. We proposed a deep CNN architecture for the blurry-vs.-sharp image classification task. We showed how weak-supervision can be employed to enable training deep neural networks in this domain as a remedy against the lack of data for blur assessment and blur localization. We compared our results with the classical methods found in the literature. We used four datasets with natural blur, three of which are sign language datasets and the remaining one was an action recognition dataset. Our results show that we can achieve near-ideal performance with a little fine-tuning on the artificial data.

Future work may focus on the use of blur as a motion-feature for other tasks; or on the effects of detecting and restoring blurry frames in SL videos for SLR performance.

Throughout this thesis we exclusively work with image based methods. That is, we consider methods that do not require any temporal information. Similar to the other works from the literature, we do not utilize information from previous and next frames while trying to estimate the middle frames' level of blur. However, it may be possible to utilize a system that uses temporal information to measure the amount of blur the individual frames have and this information may be used to train models that work on images themselves. We leave that study as a future work.

# REFERENCES


1. E01, *"London Bus"*, 2008, `https://www.flickr.com/photos/10158179@N06/2334039881`, Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0), Last Accessed 13 December 2020.

2. InspiredImages, 2015, `https://pixabay.com/images/id-843276/`, Pixabay License, Last Accessed 13 December 2020.

3. Carloyuen, 2017, `https://pixabay.com/images/id-2517653/`, Pixabay License, Last Accessed 13 December 2020.

4. Shi, J., L. Xu and J. Jia, "Discriminative Blur Detection Features", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2965–2972, 2014.

5. Zhang, S., P. Li, X. Xu, L. Li and C.-C. Chang, "No-Reference Image Blur Assessment Based on Response Function of Singular Values", *Symmetry*, Vol. 10, No. 8, p. 304, 2018.

6. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 1, pp. 172–186, 2019.

7. Marziliano, P., F. Dufaux, S. Winkler and T. Ebrahimi, "A No-reference Perceptual Blur Metric", *International Conference on Image Processing*, pp. 57–60, 2002.

8. Kalalembang, E., K. Usman and I. P. Gunawan, "DCT-Based Local Motion Blur Detection", *International Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009*, pp. 1–6, 2009.

9. Chen, X., J. Yang, Q. Wu and J. Zhao, "Motion Blur Detection Based on Low-

est Directional High-Frequency Energy", *2010 IEEE International Conference on Image Processing*, pp. 2533–2536, 2010.

10. Purohit, K., A. B. Shah and A. Rajagopalan, "Learning Based Single Image Blur Detection and Segmentation", *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2202–2206, 2018.

11. Kim, B., H. Son, S.-J. Park, S. Cho and S. Lee, "Defocus and Motion Blur Detection with Deep Contextual Features", *Computer Graphics Forum*, 7, pp. 277–288, 2018.

12. Simonyan, K. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv preprint arXiv:1409.1556*, 2014.

13. Ronneberger, O., P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

14. Zhao, W., F. Zhao, D. Wang and H. Lu, "Defocus Blur Detection via Multi-Stream Bottom-Top-Bottom Fully Convolutional Network", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

15. Tang, C., X. Zhu, X. Liu, L. Wang and A. Zomaya, "Defusionnet: Defocus Blur Detection via Recurrently Fusing and Refining Multi-Scale Deep Features", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2700–2709, 2019.

16. Qian, M., M. Xia, C. Sun, Z. Wang and L. Weng, "Defocus Blur Detection via Salient Region Detection Prior", *arXiv preprint arXiv:2011.09677*, 2020.

17. Wang, T.-L., K.-Y. Lee and Y.-C. F. Wang, "Partial Image Blur Detection and Segmentation From a Single Snapshot", *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1907–1911, 2017.

18. Ali, U. and M. T. Mahmood, "Analysis of Blur Measure Operators for Single Image Blur Segmentation", *Applied Sciences*, Vol. 8, No. 5, 2018.

19. Wang, X., X. Liang, J. Zheng and H. Zhou, "Fast Detection and Segmentation of Partial Image Blur Based on Discrete Walsh–Hadamard Transform", *Signal Processing: Image Communication*, Vol. 70, pp. 47–56, 2019.

20. Shen, A., H. Dong, K. Wang, Y. Kong, J. Wu and H. Shu, "Automatic Extraction of Blur Regions on a Single Image Based on Semantic Segmentation", *IEEE Access*, Vol. 8, pp. 44867–44878, 2020.

21. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

22. Wang, R., W. Li, R. Li and L. Zhang, "Automatic Blur Type Classification via Ensemble SVM", *Signal Processing: Image Communication*, Vol. 71, pp. 24–35, 2019.

23. Wang, R., W. Li, R. Qin and J. Wu, "Blur Image Classification Based On Deep Learning", *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6, 2017.

24. Liu, L., J. Gong, H. Huang and Q. Sang, "Blind Image Blur Metric Based on Orientation-Aware Local Patterns", *Signal Processing: Image Communication*, Vol. 80, p. 115654, 2020.

25. Yu, S., S. Wu, L. Wang, F. Jiang, Y. Xie and L. Li, "A Shallow Convolutional Neural Network for Blind Image Sharpness Assessment", *PlOS One*, Vol. 12, No. 5, p. e0176632, 2017.

26. Sun, L., S. Cho, J. Wang and J. Hays, "Edge-Based Blur Kernel Estimation Using Patch Priors", *IEEE International Conference on Computational Photography*

*(ICCP)*, pp. 1–8, 2013.

27. Xu, X., J. Pan, Y.-J. Zhang and M.-H. Yang, "Motion Blur Kernel Estimation via Deep Learning", *IEEE Transactions on Image Processing*, Vol. 27, No. 1, pp. 194–205, 2017.

28. Liang, J., G. Sun, K. Zhang, L. Van Gool and R. Timofte, "Mutual Affine Network for Spatially Variant Kernel Estimation in Blind Image Super-Resolution", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4096–4105, 2021.

29. Carbajal, G., P. Vitoria, M. Delbracio, P. Musé and J. Lezama, "Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition", *arXiv preprint arXiv:2102.01026*, 2021.

30. Pan, L., R. Hartley, M. Liu and Y. Dai, "Phase-Only Image Based Kernel Estimation for Single Image Blind Deblurring", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6034–6043, 2019.

31. Kupyn, O., T. Martyniuk, J. Wu and Z. Wang, "Deblurgan-v2: Deblurring (Orders-of-Magnitude) Faster and Better", *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8878–8887, 2019.

32. Tao, X., H. Gao, X. Shen, J. Wang and J. Jia, "Scale-Recurrent Network for Deep Image Deblurring", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8174–8182, 2018.

33. Zhang, K., W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu and H. Li, "Deblurring by Realistic Blurring", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2737–2746, 2020.

34. Liu, P., J. Janai, M. Pollefeys, T. Sattler and A. Geiger, "Self-Supervised Linear Motion Deblurring", *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp.

2475–2482, 2020.

35. Zhang, F., X. Zhu and M. Ye, "Fast Human Pose Estimation", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

36. Tran, P., A. T. Tran, Q. Phung and M. Hoai, "Explore Image Deblurring via Encoded Blur Kernel Space", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11956–11965, 2021.

37. Sultana, F., A. Sufian and P. Dutta, "Advancements in Image Classification Using Convolutional Neural Network", *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, pp. 122–129, 2018.

38. Szegedy, C., S. Ioffe, V. Vanhoucke and A. A. Alemi, "Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning", *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.

39. Chollet, F. and Others, *"Keras"*, 2015, `https://github.com/fchollet/keras`.

40. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, Vol. 15, No. 56, pp. 1929–1958, 2014.

41. Tompson, J., R. Goroshin, A. Jain, Y. LeCun and C. Bregler, "Efficient Object Localization Using Convolutional Networks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 648–656, 2015.

42. Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "Imagenet: A Large-Scale Hierarchical Image Database", *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

43. Zhou, B., A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, “Learning Deep Features For Discriminative Localization”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

44. Cheng, G., J. Yang, D. Gao, L. Guo and J. Han, “High-Quality Proposals for Weakly Supervised Object Detection”, *IEEE Transactions on Image Processing*, Vol. 29, pp. 5794–5804, 2020.

45. Choe, J., S. J. Oh, S. Lee, S. Chun, Z. Akata and H. Shim, “Evaluating Weakly Supervised Object Localization Methods Right”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3133–3142, 2020.

46. Camgöz, N. C., A. A. Kındıroğlu, S. Karabüklü, M. Kelepir, A. S. Özsoy and L. Akarun, “BosphorusSign: a Turkish Sign Language Recognition Corpus in Health and Finance Domains”, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pp. 1383–1388, 2016.

47. Ünlü, G. E., *Spatially Varying Single Image Deblurring Using Cyclegans*, Master’s Thesis, Boğaziçi University, 2020.

48. Koller, O., J. Forster and H. Ney, “Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers”, *Computer Vision and Image Understanding*, Vol. 141, pp. 108–125, 2015.

49. Konrad, R., T. Hanke, G. Langer, D. Blanck, J. Bleicken, I. Hofmann, O. Jeziorski, L. König, S. König, R. Nishio, A. Regen, U. Salden, S. Wagner and S. Worseck, *“MEINE DGS – Annotiert. Öffentliches Korpus der Deutschen Gebärdensprache, 2. Release / MY DGS – Annotated. Public Corpus of German Sign Language, 2nd Release”*, 2019.

50. Liu, C., Y. Hu, Y. Li, S. Song and J. Liu, “PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding”, *arXiv preprint*

*arXiv:1703.07475*, 2017.

51. Su, B., S. Lu and C. L. Tan, "Blurred Image Region Detection and Classification", *Proceedings of the 19th ACM International Conference on Multimedia*, pp. 1397–1400, 2011.

52. Vu, C. T., T. D. Phan and D. M. Chandler, "$S_3$: A Spectral and Spatial Measure of Local Perceived Sharpness in Natural Images", *IEEE Transactions on Image Processing*, Vol. 21, No. 3, pp. 934–945, 2011.

53. Golestaneh, S. A. and L. J. Karam, "Spatially-Varying Blur Detection Based on Multiscale Fused and Sorted Transform Coefficients of Gradient Magnitudes.", *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 596–605, 2017.

54. Yi, X. and M. Eramian, "LBP-Based Segmentation of Defocus Blur", *IEEE Transactions on Image Processing*, Vol. 25, No. 4, pp. 1626–1638, 2016.

# APPENDIX A: PERMISSIONS FOR IMAGE USAGE FROM THE LITERATURE

In the literature summary part of this thesis, Figure 2.2 is taken from [4]. The paper is published with the doi: `https://doi.org/10.1109/CVPR.2014.379`. There, the conditions for reusing the material is clearly stated as,

“Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

(i) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.

(ii) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

(iii) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.”

Our use case is in the second category and we fulfill the obligations accordingly.

Again, in the literature summary part of this thesis, Figure 2.3 is taken from [5]. The paper is published with the doi `https://doi.org/10.3390/sym10080304`. In the page it is clearly stated that the work there is “open access”, in quote: “All articles published by MDPI are made immediately available worldwide under an open access license. No special permission is required to reuse all or part of the article published by MDPI, including figures and tables. For articles published under an open access Creative Common CC BY license, any part of the article may be reused without permission provided that the original article is clearly cited." We have fulfilled this obligation by properly citing the paper.