Motion and out-of-focus blurriness: Identification and quantification

NIST PSCR Challenge 2020

Meryll Dindin, Oskar Radermecker and Pierre-Louis Missler

Team Lead
Jeffrey Hall

Partner Company
CalAster, Inc.

Location
Berkeley, California, USA

Contact
jeffrey.hall2@gmail.com

Abstract

Every 911 call goes through trained telecommunicators before first responders are sent to tackle the emergency. Telecommunicators are the first to interact with the victim, and their role is crucial in helping save lives. The significant technological improvements that the telecommunications industry has seen in the past 20 years have not reached 911 operators. Regulations, lack of incentives for innovation, and the cost of overhauling existing infrastructure have prevented 911 call centers from keeping up to date with their times. Next Generation 911 (NG911) is a government-driven initiative designed to modernize 911 call centers. The challenge: seamlessly integrate new technologies and tools while minimizing the overhead for operators, or better yet, reducing overhead by automating tasks.

Computer vision has recently reached the world of first responders, with applications ranging from body cam, drones, or even closed-circuit television (CCTV) analysis. Yet, the efficacy of these algorithms strongly depends upon the quality of the received image. A typical image impairment that first responders have to face is blurriness. Blurriness finds its cause in the fast-paced environment of most emergencies combined with the mobility of the hardware. In this report, we will be describing a relationship between blurriness and the performances of object detection models and pave the way for no-reference metrics to account for artifacts such as (1) out-of-focus objects, (2) objects in movement, and (3) camera in movement. This paper will expose how we built a specific methodology to account for those three different types of blur and quantify the level of blur in a particular image. We will then depict how we exploit state-of-the-art and open-source computer vision models for object detection and link their confidence intervals to the initial image evaluation. This methodology will, we believe, directly alleviate a huge pain point that will soon hit NG911 countrywide.

To validate our metric, we have created a dataset and assessed the efficacy of our failure rate. The dataset has been created using phone-based cameras to replicate the real-life conditions of most 911 callers. This dataset is mainly constituted of pictures taken from cameras integrated into various cellphones in various environmental conditions and on different continents. The underlying methodology was to recreate scenarios where at least one of the three types of blurs exposed above is represented. Ground truth is captured (stabilized and in-focus complementary image).

Contents

1. Main Project

1.1 Background

As Next Generation 911 becomes more widespread, Public Safety Answering Points (PSAPs) worldwide will receive not only calls but also texts and more complex pieces of media such as images or videos. Sharing pictures on the fly is an essential way for victims to help telecommunicators better understand their situation and get the best possible help. However, due to the reality of these emergencies, victims often have to act quickly. Some might be fearing for their lives, while some might be worried about others. In most case scenarios, the caller needs to split their time between facing their emergency and explaining to the telecommunicator what is happening. Due to this unique situation and because of the lack of time that callers have to review their shots, it is common for PSAPs to receive a set of images containing both sharp and impaired versions of similar scenes. As discussed in the introduction, this adds significant overhead to telecommunicators and increases the reliance on computer vision-based methods of pre-extracting information. Having an intuitive way of filtering out images from which information about the emergency cannot be reliably extracted is becoming critical.

This report will present our approach regarding the estimation of a blur as a valid artifact to computer vision algorithms in NG911. We will start by introducing our dataset and our methodology to make it relevant to the world of emergency and 911 telecommunication. We will then present our annotation pipeline and exploit SuperAnnotate and the four people working on this project to maximize standardization and minimize human errors. Following up, we will go over the computer vision use-case we selected, including the models we used as test subjects. Those will ultimately lead to our results and the establishment of relationships between the performances of object detection algorithms, blur, and no-reference metrics.

1.2 Dataset

1.2.1 Overview

CalAster decided to focus its efforts on improving the understanding of accidental blurs on incoming images shared with PSAPs. We have created a unique dataset of 528 different scenes for 1490 individual images that focus on blurriness impairments. It contains a variety of scenes taken in various lighting conditions, environments, and even continents. This dataset also comes fully annotated, reviewed, as well as with manually contoured bounding box labels. More information on the details can be found below.

The most crucial aspect for us was to create a dataset with first responders and victims in mind. Emergency callers face difficult and traumatizing situations yet still need to take photos on the spot with what is available and share them with PSAPs with a mobile device. For this reason, all scenes were exclusively photographed with a modern mobile phone to reproduce the unprepared situation that callers are in. Additionally, we wanted to have a dataset that realistically represents reality. For this reason, all impairments were naturally created with no artificial augmentation involved. This not only

guarantees that the augmentation process introduces no artifacts but it also ensures that we are working with realistic image impairments that could very well be sent to a PSAP in the case of a real emergency.

The dataset was constructed around the two most common types of blur faced by telecommunicators when receiving images: *out-of-focus blurs*, and *camera motion blurs*. The latter impairment appears particularly in poor lighting conditions when the camera aperture needs to be open for a more extended period to capture enough light to produce a sharp, discernible image. The dataset is then divided into *scenes*. A scene is a photography of the world taken in three different versions: (1) sharp, (2) out-of-focus, and (3) with camera motion. The dataset was created in such a way to allow us to compare the information content of each sharp version to its blurry counterparts. This allows us to assess the amount of information lost in the blurriness process and develop robust no-reference metrics for the assessment of blurriness down the line. All images in the dataset were acquired with a resolution of 72 dpi.

An example of a scene is provided below:

(a) Sharp image (b) Out-of-focus image (c) Camera-motion image

Figure 1.1: Example of the three different versions of the same scene.

Note Despite initially considering also exploring object-motion blurs, we learned that this type of blur was less relevant and common in this context and decided to ignore it.

1.2.2 Impairment Acquisition

The sharp and blurred versions of each scene were created based on a strict capture process applied by all photographers. Namely, the images were taken as follows:

- **Sharp images:** by making sure that the objective is focused on the scene of interest and waiting for long enough to capture a sharp representation of the scene.
- **Out of focus images:** by placing an object close to the camera to initiate an erroneous focus. This object is then removed in a quick movement, and the picture of the scene is taken.
- **Camera motion images:** by actively moving the camera during the photography process.

1.2.3 Metadata Annotation

To produce a dataset with the most value possible for future research, the team manually annotated and reviewed additional metadata for each scene in the dataset. Each annotation provides an additional piece of information that researchers can use. The following fields were annotated:

- **Mobile camera:** the mobile phone that was used to capture the scene. In total, 6 different mobile phone cameras were involved in the dataset creation.
- **Exterior or Interior:** whether interior or exterior scene.
- **Period of the day** [only for exterior scenes]: the time of day at which the scene was captured (day, night, dusk, or dawn).
- **Weather** [only for exterior scenes]: The visible weather. One of clear, cloudy, or rainy. If the sky was visibly blue or rays of sunshine were visible, the image was annotated as clear. If rain or a wet environment were visible, the image was annotated as rainy. When clouds and/or a low luminosity were present, the scene was set to cloudy. Images for which the weather was not visible received a N/A label.
- **Impairment intensity:** for each impaired image, a binary blurriness intensity score was attributed by a single annotator to decrease variability. This score can take one of two values: soft or hard.

This additional information also demonstrates the wide variety of scenes that are captured in the dataset.

1.2.4 Object Labeling

Last but not least, four annotators manually labeled more than 7800 bounding boxes of 74 different object classes on all 1489 images in the dataset.

The labeling process was performed by splitting individual images in the dataset randomly among all four annotators. Note that images, rather than scenes, were divided among annotators. This ensured a statistically unbiased labeling process where sharp images would not consistently influence the interpretation of blurry images by annotators. Having access to the sharp version of each image would lead the annotators to inject prior knowledge from the sharp version and lead to an unfair comparison with object detection algorithms. This could happen, for example, by knowing that a colorful spot in the background is a car, despite it being unrecognizable on the blurry image.

Each image was labeled using version 1.10 of SuperAnnotate by a single annotator. Figure 2.4 shows an example of an image being labeled in SuperAnnotate. Once annotated, classes and pixel definitions of the bounding boxes were extracted and saved in a separate metadata file (metadata.csv in uploaded dataset). All bounding boxes are saved in the following format: (x1, y1, x2, y2) where x1, y1 corresponds to the top-left pixel of the bounding box, whereas x2, y2 corresponds to the bottom right pixel. Definitions of edgy individual classes were continuously discussed internally for increased consistency (e.g., deciding whether a kayak is a boat or determining whether a van is a bus, a truck, or a car). The final labeled dataset was eventually reviewed and corrected by a single annotator to increase the label consistency further, fix any remaining errors, and remove inter-observer variability.

In total, 90 classes of objects were annotated in the dataset, including high importance labels such as person, car, and street sign. Classes are annotated consistently with the famous COCO dataset Lin et al. (2014) on which most modern state-of-the-art object detection algorithms are trained, including Yolo, Retinanet, and Facebook's Detectron. Because trained human annotators manually drew these bounding boxes, they represent the maximum amount of information that can be extracted from individual images for object detection.

These labels have two primary purposes. First, they allow us to directly assess the reduction of information due to blurriness in each scene and correlate that reduction to the intensity of blurriness computed via non-reference metrics, thereby supporting further research in the field. Second, they allow us to benchmark the failure rates of the most modern object detection models on both sharp and impaired

Figure 1.2: Annotation of an image from the dataset in SuperAnnotate 1.10.

versions of the same scene, thereby quantifying how these impairments affect automated information extraction methods that can be used in PSAPs to support telecommunicators.

1.3 Object Detection

To assess the impact of the image impairments, we applied three state-of-the-art object detection algorithms on all our images in inference: Retinanet, Yolov4, and Detectron2.

An overview of the number of objects detected per model and per image impairment category:

Number of detected objects	Sharp	Out of focus	Camera motion
Retinanet	2012	1153	785
Yolo v4	4046	2033	1654
Detectron v2	4392	2291	1853
Manual Annotation	4007	1993	1875

Table 1.1: Number of objects detected per model and per camera impairment category.

1.4 Failure Rate

To estimate the impact of out-of-focus and camera motion-induced blurs on object detection models, we calculated several state-of-the-art metrics used by researchers in the computer vision field. More specifically, we compared the Average Precision (AP) and Average Recall (AR) between manually annotated

and predicted bounding boxes for sharp and impaired images independently.

To understand these metrics, let's first define precision and recall. They are given by the following formulas:

$$Precision = \frac{TP}{TP + FP} \tag{1.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{1.2}$$

where TP are the true positive, FP the false positive, and FN the false negatives.

Precision represents the portion of correctly detected objects in an image out of all the detected things by the model. On the other hand, recall represents the portion of correctly detected objects out of all the actual objects in the ground truth.

However, the above formulas assume that we know the bounding boxes. Yet, the model predicts objects with a certain probability. Setting the prediction threshold to a fixed value, such as 0.5, is a good start but can introduce bias in the score. An Average score, therefore, considers the average of its value over all possible cut-off thresholds.

Given these clarifications, it is easy to see that precision and recall are the metrics that will be of most interest in our specific situation. When an image is blurry, the object becomes unrecognizable and will not be detected. This can be confirmed by looking at the number of detected objects in the 'out of focus' and 'camera motion' columns as compared to the 'sharp' column above. As a consequence, precision can lead to artificially higher scores simply because the model on heavily blurred images makes very few predictions. On the other hand, recall is bound to the manual ground truth labels, which are annotated with a deeper understanding of the context in which these objects evolve. Those two metrics are thus complementary and are sufficient to assess object detection model performances.

This being stated, the metrics we went for are **Average Precision** and **Average Recall**. This notion of average is due to the inherent behavior of recall and precision. Recall values monotonously increase as we add more predictions, while precision has more of a zigzag pattern as it goes down with false positives and goes up again with true positives. Ultimately, Average Precision (AP) gets defined as the area under the precision-recall curve computed on cumulative predictions. To be even more precise, the metric we use is mAP (mean Average Precision), defined as the average of AP, but in our context (COCO context), they mean the same thing.

One will see that our graphs also contain other metrics, that we similarly used during our exploration. Those are scores across scales or Intersection over Union dependent. Intersection over Union (IoU) is a popular metric in object detection since the Pascal VOC challenge, popular in computer vision. It simply is a useful metric characterizing how close a predicted bounding box is to the ground truth annotation. The associated formula is reduced to the ratio in between the area of overlap and the area of union between the predicted bounding and the ground-truth. Stemming from those metrics, here are a few handful other metrics we used in exploration, without going in too much details for the sake of this challenge:

AP50	average AP for IoU from 0.5 to 0.95
AP75	average AP for IoU exactly equal to 0.75
APlarge	evaluates the ground-truth objects of large ground-truth objects (area above 96x96)
AR1	AR given 1 detection per image
AR10	AR given 10 detection per image
AR100	AR given 100 detection per image
ARlarge	evaluates the ground-truth objects of large ground-truth objects (area above 96x96)

Table 1.2: Other metrics used during exploration for a more indicative trend observation.

1.5 Impact of Blurriness

While building our dataset, our protocol was about ensuring scene consistencies while emulating the two types of blur we were looking for: motion blur and out-of-focus blur. This construction of our dataset enables us to compare a relatively similar scene through three lenses: a sharp one and two blurry ones. Our experiment then consisted in running the three object detection models introduced above and compare our manual annotations to the model predictions. As expected, blur resulted in a degrading impact over our model performances of 7% in average for camera motion and 3% in average for out-of-focus blur.

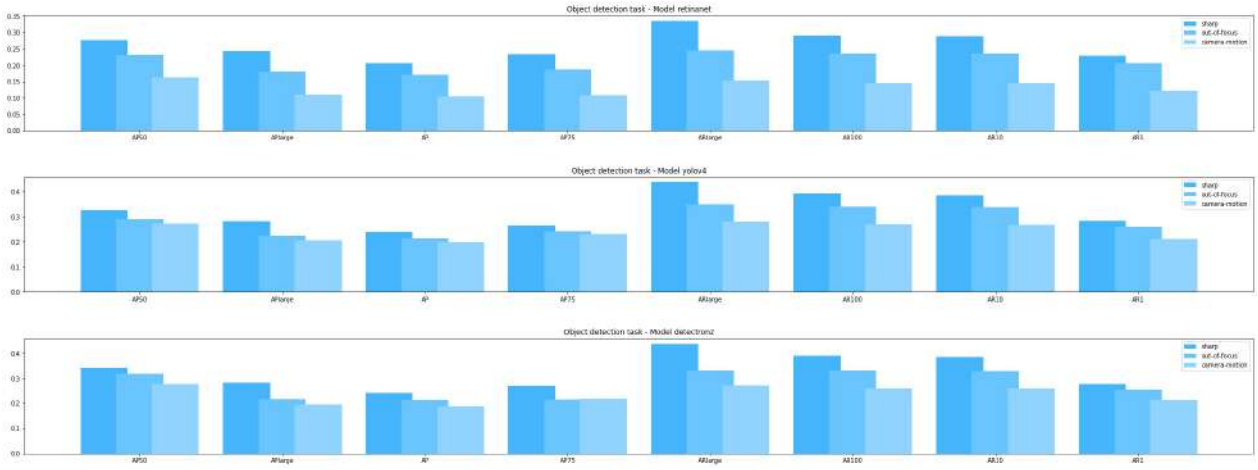

Figure 1.3: Failure rates on our dataset, using our manual annotations as ground truths

This perfectly illustrates the importance of blur monitoring in computer vision applications and the usefulness of the selected failure rate metrics. Our goal is to ultimately prepare the next wave of AI-powered tools for telecommunicators; those results show the importance of filtering content such as videos and images for first responders while being aware that image artifacts considerably reduce your confidence interval correctly perform such task.

	Model	AP50	APlarge	AP	AP75	ARlarge	AR100	AR10	AR1
sharp	yolov4	0.326	0.281	0.239	0.264	0.437	0.391	0.384	0.284
out-of-focus	yolov4	0.292	0.223	0.212	0.242	0.349	0.34	0.337	0.258
camera-motion	yolov4	0.272	0.205	0.197	0.23	0.28	0.269	0.268	0.21
sharp	detectron2	0.341	0.283	0.242	0.269	0.438	0.391	0.386	0.278
out-of-focus	detectron2	0.318	0.216	0.212	0.214	0.333	0.331	0.328	0.254
camera-motion	detectron2	0.277	0.195	0.187	0.218	0.27	0.259	0.258	0.211
sharp	retinanet	0.277	0.243	0.205	0.235	0.336	0.291	0.289	0.228
out-of-focus	retinanet	0.232	0.18	0.17	0.187	0.245	0.235	0.235	0.206
camera-motion	retinanet	0.163	0.111	0.106	0.109	0.154	0.146	0.146	0.122

Table 1.3: Computed metrics on our crafted dataset.

1.6 No-reference Metrics

The previous experiment would not be complete without linking possible no-reference metrics to the outcomes we presented. We spent time reviewing literature and came up with a few no-reference metrics we deemed useful in our context. Ultimately, we would want to establish a relationship between any of those metrics (or one of their combination) and the actual outcome of our object detection models. The ones we cherry-picked are listed below:

- **Laplacian Variance** (Bansal et al. (2016))
- **Tenengrad Variance** (Huang and Jing (2007))
- **Fast Fourier Transform** (Hassen et al. (2013))
- **Wavelet Transform** (Vu and Chandler (2012))
- **Discrete Cosine Transform** (Jiao et al. (2014))
- **Sobel Filtering** (Varadarajan and Karam (2008))

The first question that stemmed from the metrics computation was whether we would see any trend or correlation between the simple categorical classification of our images (sharp vs camera-motion vs out-of-focus) and the no-reference metrics. To answer such a question, we computed all the scores presented above, calculated their respective Pearson correlation in between each other, and reorganized them by hierarchical clustering. Overall, two elements are to be extracted from the visualization below: (1) the image size is correlated with none of those measures, which is essential for any regression task with no-reference metric (2) the metrics are all positively correlated with sharpness and negatively correlated with blur. Lacking granular levels of blur in our dataset, we will, unfortunately, be unable to analyze this trend with those observations further. Nonetheless, that analysis tells us that either the Discrete Cosine Transform or the Fast-Fourier Transform would be very promising no-reference metrics to monitor blur (since both are also very correlated, it would not be necessary to use both).

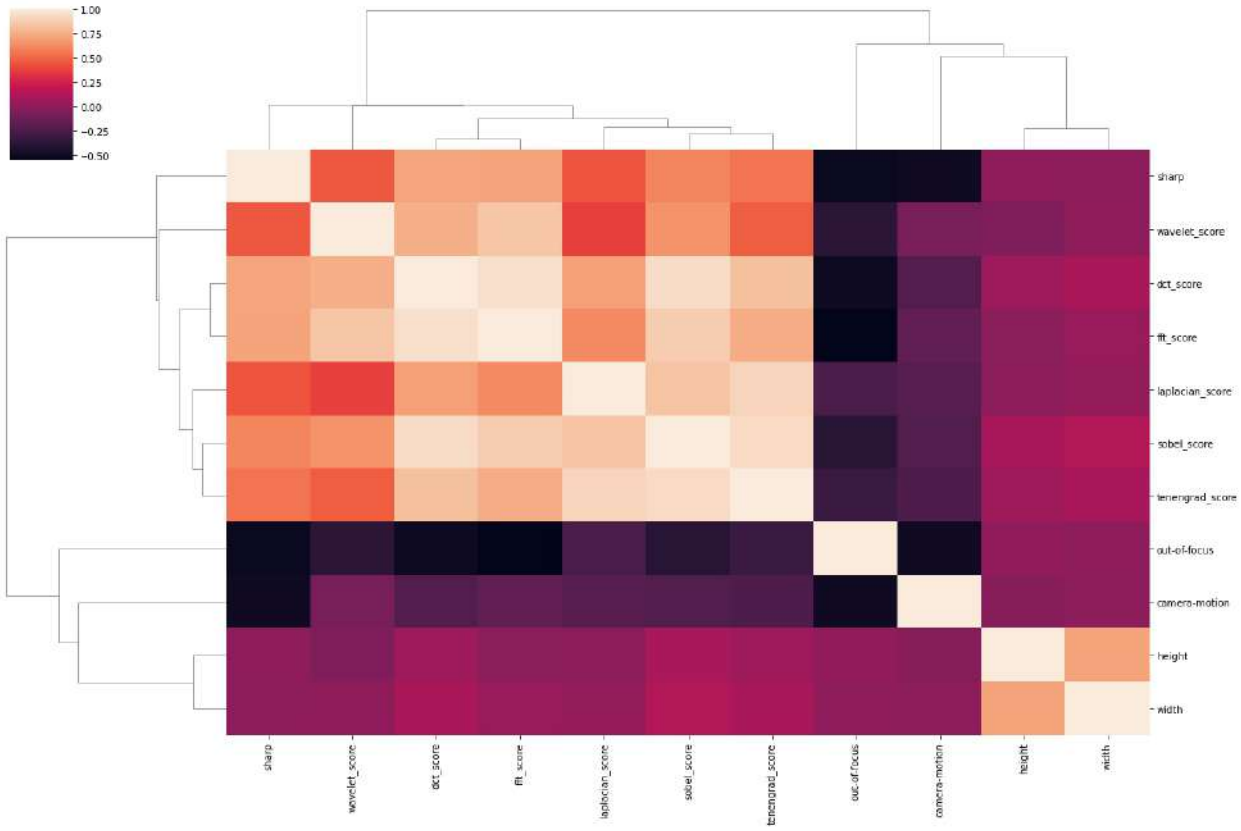

Figure 1.4: Hierarchically clustered correlations of our no-reference metrics with our labels (sharp, camera-motion, out-of-focus)

	sharp	out-of-focus	camera-motion
Discrete Cosine Transform	0.723	-0.504	-0.221
Fast-Fourier Transform	0.718	-0.556	-0.163
Laplacian Variance	0.446	-0.251	-0.196
Sobel Filtering	0.61	-0.386	-0.226
Tenengrad Variance	0.55	-0.315	-0.237
Wavelet Transform	0.457	-0.377	-0.081

Table 1.4: Pearson correlations in between no-reference metrics and our blur categories.

2. Results of our Artificial Dataset

2.1 Overview

To account for some of the challenges we faced while building our dataset, we prepared an experimental pipeline to test out no-reference metrics and our failure rates robustness. Among those challenges, we find the unreliability of scene superposition across images taken manually (e.g., a car in a sharp image would not be at the same position in our blurry take of that same scene) and the inherently limited scope of photos we can take as pedestrians. This was, for us, an excellent way to augment our approach through a controlled experiment where we could minimize the variation of external factors. It also gave us a way to fine-tune the level of artificial blur and thus offer us a more granular visualization of the impact of blur on the performances of object detection models. In this section, we will go through our process for image collection, our methodology for artificial blur, the selection of our object detection models, and the resulting failure rates and no-reference metrics.

As per the copyright regulations and the challenge guidelines, this dataset is not released nor included in our submission.

2.2 Image Scraping

It begins with image collection. Nowadays, a quick and reliable way of finding content is to exploit search engines, as would anyone do by typing specific keywords in a search bar on google.com. To entertain this idea of a controlled experiment, we cherry-picked the labels we deemed the most interesting in our use case: cars and people. The thinking behind this limited selection originates from our 911 use-case crossing the performances of object detection models, which generally perform better on those labels (mainly because of data availability). The 911 use-case we refer to are car accidents and people interactions on emergency scenes. Using an emulated Chrome driver, we could utilize the Google search engine to retrieve as many images as we wanted for those two labels. The code we used was inspired by this Github repository. For anyone trying to reproduce this experiment, here are the keywords used for the search: *car accident*, *car highway*, *people running*, *people talking*, *people walking* and *traffic jam*. We limited the scraping to 2600 pictures.

2.3 Artificial Blur

Following up on this notion of controlled experiment, we relied on artificial blur to recreate scenes applicable to showcase our failure rates. Because all blurs we used depend on the idea of kernel size, we could easily parametrize and uniformize the amount of blur in each artificial image. Our overall rule was to sample 11 levels of a blur for each image, computed as a uniform sampling in between 5 pixels and 20% of the minimum in between the image height and image width. Blur levels would thus be consistent across images, independently from their resolutions.

2.3.1 Out-of-focus Blur

Out-of-focus is a common artifact associated with taking photos. When a camera images a real scene, some of the points are in focus while others not, thus causing out-of-focus blurring. Creating an out-of-focus blur is nothing more than convolving an image with a gaussian function. The kernel size of that function will dictate the level of artificial blur in the image. In terms of image processing, any sharp edges are smoothed while minimizing too much blurring: applying a Gaussian blur has the effect of reducing the image's high-frequency components and thus acts as a low pass filter. An example of such blur generation is provided below, presenting every one of the 11 levels of blur we went for.

Figure 2.1: Example of an artificially blurred image to resemble out-of-focus blur

2.3.2 Motion Blur

Motion blur is the streak-like effect of taking a photo while rushing, or when the camera exposure is particularly long (i.e., time-lapse photography). In our case, this is representative of emergencies when a person is under pressure and needs to flee while taking the picture. Applying motion blur to an image consists of filter convolution across the image. We can make it happen horizontally, vertically, or in any direction we want. Technically speaking, a motion blur kernel averages the pixel values in a particular direction and acts as a directional low pass filter.

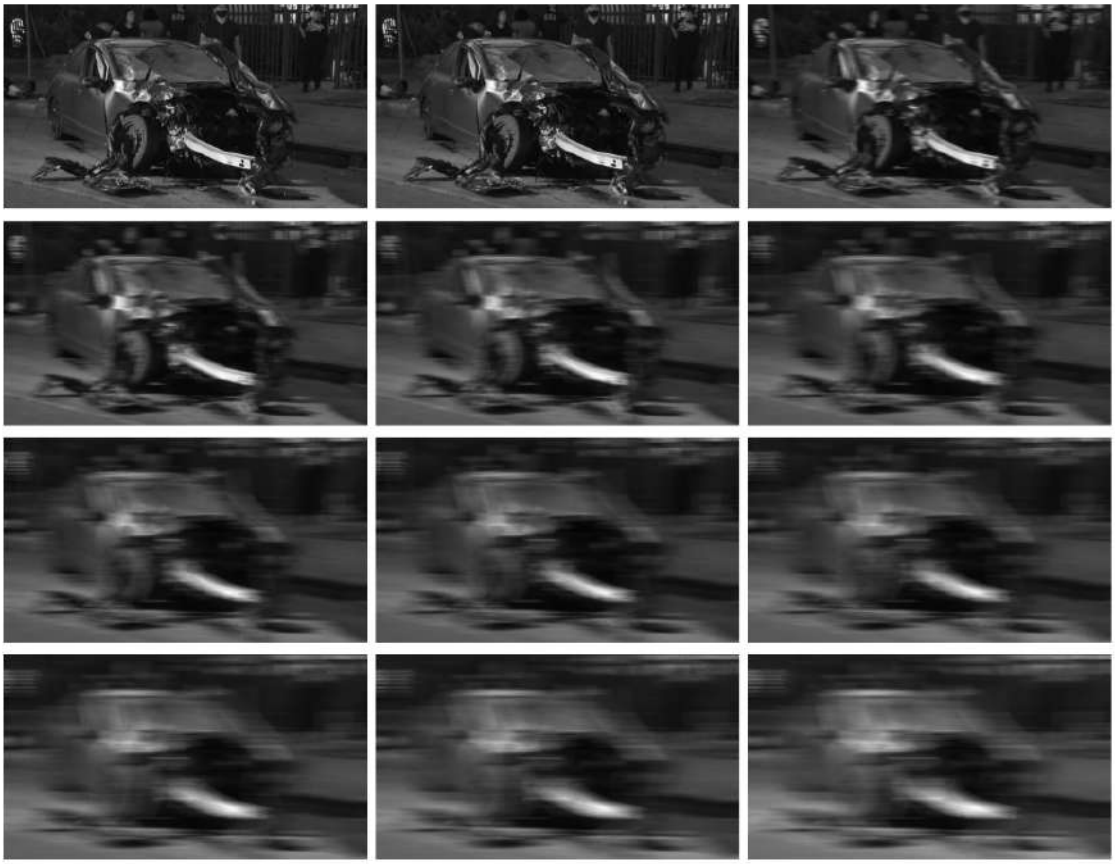

Figure 2.2: Example of an artificially blurred image to resemble motion blur

2.4 Object Detection

To assess the impact of the image impairments, we applied two state-of-the-art object detection algorithms on all our images in inference: Retinanet and Detectron2, similarly to our natural dataset. An overview of the number of objects detected per model on our scraped dataset is given here:

	Retinanet	**Detectron v2**
Number of detected objects	19648	32961

Table 2.1: Number of objects detected per model.

2.5 Failure Rates

To determine the impact of blurriness on our selected failure rates, we ran the controlled experiment depicted above by artificially augmenting blur on our scraped images. This resulted in a granular observation of performances of our object detection models, as presented below.

For the sake of time and because this experiment is out of the scope of this challenge, we did not go through the hurdle of manually annotating our artificial dataset. Instead, because images maintain a spatial relationship after being artificially enhanced, we made the strong hypothesis that our ground

truths are provided by the output of our object detection model on the sharpest image of all. By doing so, we could fully automate our pipeline to generate as many experiments as we wanted. Our approach to scoring and failure rates is the same as the one used for our natural dataset, relying on Average Precision and Average Recall. The results are consistent with what we observed for our realistic dataset, except that we now have 11 points to estimate a trend, rather than only 3. This now makes it easier to build a regression model to link it to the no-reference metrics we exposed in the previous section. The graph below confirms our expectations and our previous results showcasing that blur is an artifact degrading the performance of computer vision models.

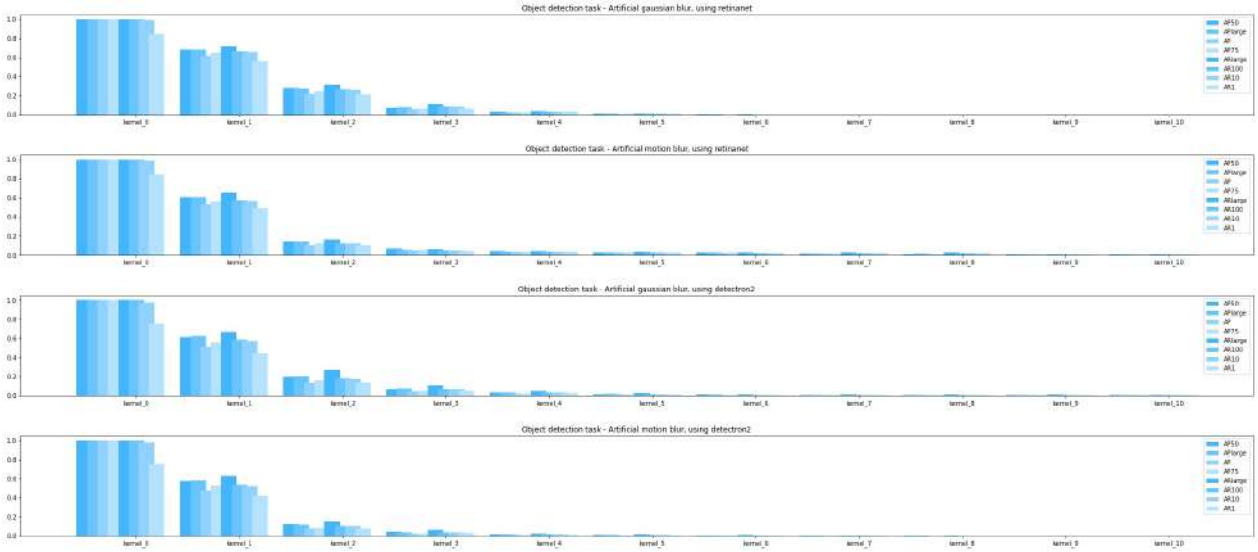

Figure 2.3: Failure rates on our multi-kernel experiment, using each model predictions on the sharpest images as ground truths

An extra step in that thought experiment was to use models as independent ground truth for another model. For example, in the graph below, we used Detectron2 predictions as the ground truth to all our images while evaluating Retinanet performances on artificially augmented images. Results remained consistent with our experiments so far, with blur influencing our model performances negatively.

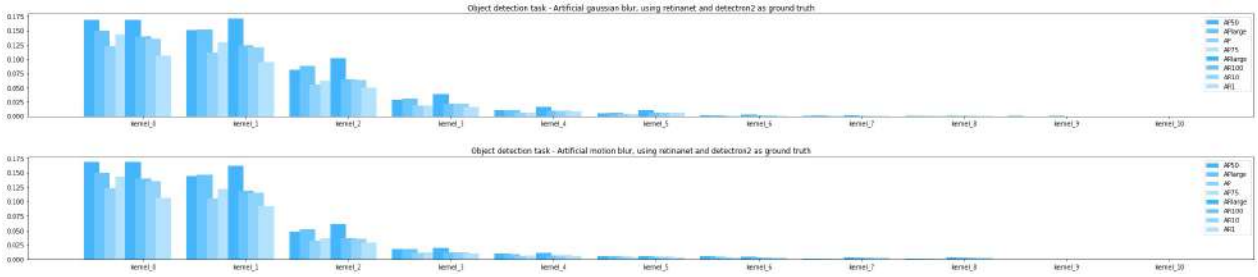

Figure 2.4: Failure rates on our multi-kernel experiment, using the Detectron2 predictions on the sharpest images as ground truths

3. Closure Thoughts

In this report, we can draw three main lessons from our experiments.

- It is effortless to take a blurry image. As we went through creating our dataset, we simulated conditions where the blur artifact could occur. This could be done because of an object attracting the automatic focus of the smartphone, passing a hand in front of the objective or simply taking the picture too fast. Similarly, for camera motion, running or turning while taking a picture was sufficient to make the picture unusable. If anything, this is a good hint that the NG911 systems will soon be overflown by images of different qualities, some that will not be usable by first responders and thus making them lose precious seconds while processing the received information. It proved to us the need to start thinking about proper architectural strategies to avoid the overwhelming of our telecommunicators.
- Blur acts as an artifact degrading the performances of object detection models. The failure rates we selected (Average Precision and Average Recall) reflect this degradation across multiple blur levels, as our natural and artificial experiments show it.
- No-reference metrics can be computed to represent a level of blur in an image. As this was not precisely the goal of this challenge, we did not want to deep dive into the relationships between the no-reference metrics we proposed. Nonetheless, we showed clear correlations between those and the simple blur levels we used as categorical labels.

We provided a starting point to account for blur as a valid computer vision artifact that will ultimately be a pressing challenge for the incoming NG911 ecosystem. Prior algorithmic and filtering decisions can be made to avoid the saturation of an already saturated group of first responders. We aimed here to pinpoint quantifiable manners to predict performance degradation of model performances, which could be achieved through the last mile study of robust no-reference metrics and model performances.

4. Our Team

Our team is composed of three data scientists that graduated from UC Berkeley. We have been strongly involved in the 911 ecosystem for two years now, and have been observing the lack of technology for first responders, and decided to help bring our knowledge and skills to help develop the field. The three of us have significant experience in computer vision, data science, data engineering, cloud computing, HIPAA regulations, as well as an increasing understanding of the constraints in a 911 call center. Below is a more in-depth description of everyone's skills and motivations.

1. **Meryll Dindin**: Meryll is a mission-driven entrepreneur, working at the intersection of artificial intelligence and social impact. Meryll has worked as a data scientist for mid-to-large companies such as UNICEF, Fujitsu, Zipline, Glooko. He holds a MSc in Computer Science, an BSc in Biomedical Engineering and a MEng in biomedical engineering and data science from UC Berkeley. Meryll most recently worked on computer vision for human phenotyping, which includes models such as face detection, emotion analysis, body pose estimation, heartbeat estimation, ... (see demo). His expertise spans from specific model construction to project management.
2. **Oskar Radermecker**: Oskar is a passionate computer vision-focused data scientist with the goal of helping develop technology for good. Oskar has worked as a data scientist for various academical institutions including UC San Francisco, UC Berkeley, and Monash University where he applied his skills to the analysis of cells on histological slices. More recently, Oskar has been working for Varian Medical systems where he is developing convolutional neural networks for the classification and segmentation of medical images. He holds a BSc in Biomedical Engineering, an MSc in Biomedical Engineering, and a MEng in biomedical engineering and data science from UC Berkeley.
3. **Pierre-Louis Missler**: Pierre-Louis Missler graduated from ESTP Paris with a MSc in Mech engineering and from UC Berkeley with a MEng in Systems in 2019. His primary focus is to improve access to new technology in the emergency sector around a tech for good company. With his data science background, he provides insights and technical vision to carry projects from A to B while ensuring that best practices are applied.

Bibliography

Bansal, R., G. Raj, and T. Choudhury (2016). Blur image detection using laplacian operator and open-cv. In *2016 International Conference System Modeling Advancement in Research Trends (SMART)*, pp. 63–67.

Hassen, R., Z. Wang, and M. M. A. Salama (2013). Image sharpness assessment based on local phase coherence. *IEEE Transactions on Image Processing 22*(7), 2798–2810.

Huang, W. and Z. Jing (2007). Evaluation of focus measures in multi-focus image fusion. *Pattern Recognition Letters 28*(4), 493–500.

Jiao, S., H. Qi, and W. Lin (2014). No-reference perceptual image sharpness index using normalized dct-based representation. In *2014 Seventh International Symposium on Computational Intelligence and Design*, Volume 2, pp. 150–153.

Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár (2014). Microsoft coco: Common objects in context. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.

Varadarajan, S. and L. J. Karam (2008). An improved perception-based no-reference objective image sharpness metric using iterative edge refinement. In *2008 15th IEEE International Conference on Image Processing*, pp. 401–404.

Vu, P. V. and D. M. Chandler (2012). A fast wavelet-based algorithm for global and local image sharpness estimation. *IEEE Signal Processing Letters 19*(7), 423–426.