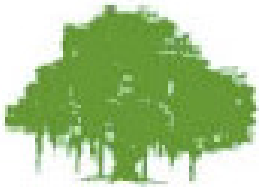
Tackling Low Resolution for Better Scene Understanding

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS in Computer Science and Engineering
By Research

by

Harish Krishna
201202172
harishkrishna.v@research.iiit.ac.in



International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Tackling Low Resolution for Better Scene Understanding" by Harish Krishna, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. C.V. Jawahar

To THE READER

Acknowledgments

I'm immensely thankful and honoured to have worked under the guidance of Prof. C.V. Jawahar a veritable reservoir of ideas, wisdom, experience, and patience. His brilliance in taking things forward when projects seem to be plateauing, innovative ideas for social connect, presentation skills, and the sheer diversity of the numerous problems he is involved in, among several other qualities and experiences make him the ideal researcher to emulate.

I would also like to acknowledge my co-guides from whom I've learned so much over the years. My gratitude to Dr. Karteek Alahari and Dr. Anand Mishra who taught me how to set up and perform experiments. I am thankful to all the faculty of the Center of Visual Information Technology, particularly Dr. Girish Varma and Dr. Anoop Namboodiri for insightful discussions on problems, technologies and quick solutions.

Last but not the least, I would like show gratitude to the entire CVIT ecosystem. As a cluster admin, I was fortunate to have got the opportunity to talk to nearly every student, and have beneficial technical discussions with each and every one of them. The regular study group meetings and frequent guest lectures have had great influence in whatever little I know now. I shall always look back at my time in the lab with very fond memories.

Abstract

Complete scene understanding has been an aspiration of computer vision since its very early days. It has applications in autonomous navigation, aerial imaging, surveillance, human-computer interaction among several other active areas of research. While many methods since the advent of deep learning have taken performance in several scene understanding tasks to respectable levels, the tasks are far from being solved. One problem that plagues scene understanding is low-resolution. Convolutional Neural Networks that achieve impressive results on high resolution struggle when confronted with low resolution because of the inability to learn hierarchical features and weakening of signal with depth. In this thesis, we study the low resolution and suggest approaches that can overcome its consequences on three popular tasks - object detection, in-the-wild face recognition, and semantic segmentation.

The popular object detectors were designed for, trained, and benchmarked on datasets that have a strong bias towards medium and large sized objects. When these methods are finetuned and tested on a dataset of small objects, they perform miserably. The most successful detection algorithms follow a two-stage pipeline: the first which quickly generates regions of interest that are likely to contain the object and the second, which classifies these proposal regions. We aim to adapt both these stages for the case of small objects; the first by modifying anchor box generation based on theoretical considerations, and the second using a simple-yet-effective super-resolution step.

Motivated by the success of being able to detect small objects, we study the problem of detecting and recognising objects with huge variations in resolution, in the problem of face recognition in semi-structured scenes. Semi-structured scenes like social settings are more challenging than regular ones: there are several more faces of vastly different scales, there are large variations in illumination, pose and expression, and the existing datasets do not capture these variations. We address the unique challenges in this setting by (i) benchmarking popular methods for the problem of face detection, and (ii) proposing a method based on resolution-specific networks to handle different scales.

Semantic segmentation is a more challenging localisation task where the goal is to assign a semantic class label to every pixel in the image. Solving such a problem is crucial for self-driving cars where we need sharper boundaries for roads, obstacles and paraphernalia. For want of a higher receptive field and a more global view of the image, CNN networks forgo resolution. This results in poor segmentation of complex boundaries, small and thin objects. We propose prefixing a super-resolution step before semantic segmentation. Through experiments, we show that a performance boost can be obtained on the popular streetview segmentation dataset, CityScapes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scene understanding has been an important goal of computer vision systems - to be able to make machines identify objects and surfaces in a picture and interpret relationships between them. It was realised that "if the identity and locality of all the objects in a scene are known, then recognition would be perfect". Scene understanding since then has evolved and lies in the core of many imminent successes of modern technology - autonomous driving, human-machine interaction, remote sensing, among several others.

Object detection is the keystone of scene understanding. The goal of object detection is to find the spatial coordinates of an unknown number of instances of specific categories in an image. Object detection has applications in robotics, remote sensing, autonomous driving, surveillance, medical imaging and nearly in every application domain of computer vision. While this problem has seen a lot of interest and successes in the deep learning era, object detection as it stands today is far from being solved. One major point of failure of even the best detectors is that they fail when trying to detect very small objects. This important problem is the focus of chapter 2 of this thesis, where we use super-resolution to enhance region proposals for better classification.

The faces in a scene can also be seen as objects that need to be detected and recognised. Though face recognition has been studied for biometrics for a long time, identifying all the faces in an unconstrained, in-the-wild scene has received surprisingly little attention. One particularly interesting scene is of semi-structured settings where the identities of the faces are known and faces are somewhat frontal. These scenes pose unexpected challenges for face detection and recognition systems which we describe in length in chapter 3.

Object detection approaches are usually content with just finding the rectangle that bounds the object that is detected. Applications might require more accurate localisation that encapsulates the shape and appearance of the object. An extreme scene understanding problem is the task of semantic segmentation where the goal is to assign a semantic class label to every single pixel of the image. Existing approaches are confused by objects that may look alike but endemic to very different scenes. To mitigate this, approaches give the network a larger field of view to better understand the scene, thereby losing out on valuable high-frequency local information. Autonomous driving now relies on semantic segmentation

to know the exact boundaries of roads and obstacles. In chapter 4, we ask if knowing the scene helps us deliver better segmentations.

The improvements we suggest to all three problems stem from the observation that the deep learning methods in vogue perform poorly for low-resolution regions of interest. In the rest of this chapter, we discuss what resolution is and what it means to have a lower resolution. We then list scenarios which can be plagued by low-resolution. We then try to understand why deep learning methods have mediocre accuracies for tasks on low-resolution while working so well for high resolution. Then we see how some recent approaches try to overcome the bane of low-resolution.

1.1 Terms and Definitions

It is essential to refresh some often-confused terms and definitions about resolution before proceeding forth. One possible reason for confusion (say, between interpolation and upsampling) is because these operations perform similar tasks, in this case, usually give us a larger version of an image. Sometimes, different research groups working independently on the same idea assign different nomenclature to the same operation. This section also tries to clarify some of these terms which would be used later in the thesis.

- **Resolution** - This is a measure of the density of pixels in a grid. It is equivalent to the count of effective pixels that contribute to an image.
- **Resizing** - This is the process of changing the size of an image while keeping the sampling rate the same. No new information is added to the image. Also called **rescaling** and **interpolation**.
- **Super-resolution** - This is the process of obtaining a high-resolution image from a low-resolution observation. It can be seen as making the sampling grid more dense by adding more pixels per unit area. Also called **upsampling**.
- **Deconvolutional layer** - This is an operation that is commonly used to learn an upsampling. Deconvolutional layers do not perform the deconvolution operation; they are normal convolutional layers with fractional strides. Also called **transpose convolution**, **sub-pixel convolution** and **backward convolution**.

1.2 The Low Resolution

It turns out that low resolution can hamper machine vision, just like it does to human vision. Before discussing "why" this warrants separate study and "how" to overcome its effects, we look at "where" we might come across low resolution in vision.

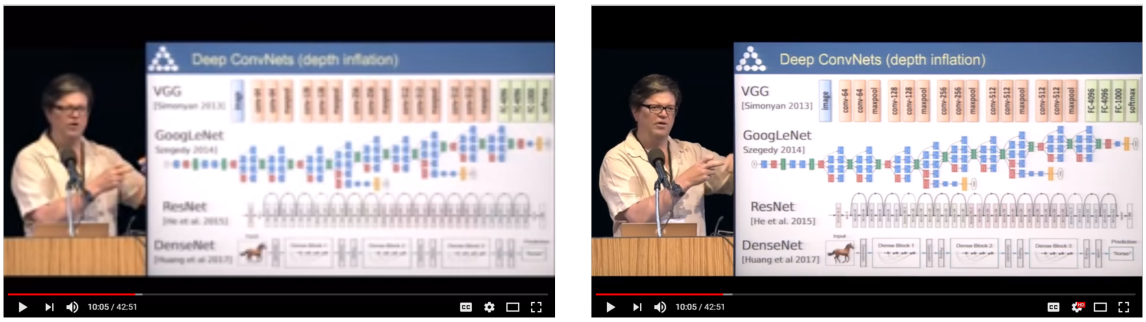


Figure 1.1 Resolution and scale are often confused. The first screenshot is from a YouTube video played at 144p resolution while the second has 720p resolution. The higher-resolution video packs more information per unit area. Notice that though it is much easier to read text and recognise faces in the higher resolution frame, it isn't impossible in the lower resolution.

The most obvious scenario where low resolution manifests is when the camera is of low resolution. Despite recent advances in cheap, light-weight and high definition cameras, people and establishments might not switch to them due to affordability, the labour involved in replacing existing cameras or lack of awareness. Another scenario is when we might have to be working on images and videos that precede this recent spurt in high-quality cameras. These apart, there is still a major interest in using a low-resolution camera, particularly in surveillance. Apart from prohibitive costs, privacy is a major concern with high-definition video cameras. If there are video cameras that monitor a house in an always-on mode, the dwellers' private lives can be viewed or recorded by a hacker who manages to tap into the feed. However, if the resolution is substantially low, faces might be unrecognizable while human activities can still be identified [68]. Low resolution surveillance also has advantages of requiring much less bandwidth (for smart-home applications) and storage space. Low resolution can also be a consequence of using a wide-angle camera. Though the image is captured in high definition, the object of interest could occupy a very small part of the image.

However, there are more subtle and unintentional scenarios where the resolution of objects might be low:

- the object of interest is small - even if the entire image is in high resolution, an object could be so small that when only considering that region, it is in low resolution. Even if the object is in the foreground and in focus, unless it is the primary subject of the photo, it might occupy a small area of the image. Objects that are commonly small in the real-world tend to be small in scene images. Another scenario is in medical imaging where tumors and abnormalities that need to be detected in scans are tiny. Low resolution makes detecting small objects a challenging task. The challenges and approaches to overcome this are discussed in detail in chapter 2.

- the object of interest is far away - another cause for an object to occupy a small area of an image is if the object is far away from the camera. The objects itself could be big such that when they

are brought out of the far-field, they occupy reasonably high resolution. Far away objects have an added disadvantage that they are out of focus and many times, not well lit. One interesting case where the differences between this case and the above is when detecting and recognising faces in a large lecture hall. Though all the faces have roughly similar sizes, those at the back are of such staggeringly low resolutions, that they become very hard for a system. This problem is introduced and analysed in chapter 3.

- trade-off for bigger field of view - resolution is applicable to feature maps and convolutional activations, and not just to natural images. Convolutional neural networks are inspired by the way humans perceive hierarchically and look at larger regions of an image with depth. Because of difficulties in training larger filters, deep CNNs increase the receptive field by downsampling the activation maps through pooling layers. The hope in intentionally removing information is that local and high-frequency information is already analysed and presence is captured in the activations produced by the shallower layers. While this trade-off works very well in practice for a task like object classification, it is costly for a task like semantic segmentation where the high-frequency and location information are essential while trying to construct segmentation masks at the original resolution of the image. The effect is pronounced for complex boundaries, thin and small objects. In chapter 4, we suggest a method which handles this loss of resolution due to strided convolution and pooling layers for the problem of semantic segmentation.

We can therefore see that we will have to deal with low resolution in problems of object detection, semantic segmentation and face detection and recognition. But why should vision on low resolution (LR) be different from vision on high resolution(HR)? After all, a low resolution and high resolution image of the same scene both contain the same objects. The rest of this section focuses on establishing that low resolution vision has its own challenges which make it a hard problem.

As we might have observed from figure 1.1, recognising faces becomes harder for humans at lower resolutions. Experiments performed in [4] and [53] confirm that several face recognition algorithms fail at low resolution. Now, if both low and high resolution faces were from the same manifold, then it shouldn't be hard for neither humans nor machines. This shows that the low resolution manifold is significantly different from the medium and high level manifold.

To accommodate that the scenes in both resolutions were the same, it was assumed [8, 76, 58] that the manifolds looked alike. More formally, it was taken that the distance between two high resolution images must have been the same as the distance between their low resolution counterparts. Interestingly, some approaches used this for super-resolution: During the train phase, create high resolution and low resolution manifolds using patches from high resolution training images and their downsampled versions respectively. For a low resolution test image, divide the image into patches, and for each patch, use the k nearest neighbours in the low resolution manifold to find the corresponding allegedly-neighbouring patches in the high resolution manifold and average them to find the mapping. Because of

the assumption that the two manifolds have similar geometrics, the locality of the LR test patch in the LR manifold would be alike the locality of the super-resolved patch in the HR manifold. However, the results of this method were ordinary and several approaches, even preceding it, performed better [16] without the assumption. Note that LR scene or face images and small patches extracted from HR images are very different, despite both technically packing information in an equivalent number of pixels. [53] showed that similarity was very different in the LR and HR spaces by probing LR images within a HR gallery. Another observation [68] which suggested that the two manifolds were different was that even small changes in pixel values had large changes on the LR images. For instance, while the HR space was robust to small translations, even sub-pixel translations resulted in significantly different LR images.

Much later, through careful empirical study of the surfaces, [83] found that the LR and HR manifolds do not overlap, even after sophisticated non-linear learned transformations. They concluded that it is best to think of the two resolutions as dissimilar domains.

Because the LR and HR spaces have little overlap or similarity in structure, it now makes sense that many of our computer vision algorithms work well for one resolution, but poorly for the other [2]. This was empirically verified by several works in the recent past [62, 4]. Apart from furthermore proving that the HR and LR spaces are different, these results highlight that learning features using deep learning for low-resolution is harder.

Peng et.al. [62] synthetically created low-resolution versions of popular datasets and experimented to find the performances on both resolutions. The following table discusses image classification for the Stanford Cars dataset using the AlexNet [41] architecture. The results are striking:

Train	**Test on High-res**	**Test on Low-res**
High	80.3	1.7
Low	13.3	50.4
High + Low	72.9	59.3

Table 1.1 Training and testing on high and low resolution versions of images from the Stanford Cars dataset by [62] clearly indicate the gaps in the two domains. It seems as if the features discriminative in the HR domain are nearly useless in the LR domain. The results on other popular image classification datasets are similar.

While the classification accuracy on high resolution images trained on high resolution images is very high, the same network performs very poorly on low resolution images. Also to be noted is how training a state-of-art CNN on low-res images gives 30% less accuracy than training and testing on high-res images. A redemption for us is that training along with high-resolution images gives an improvement over training only with low resolution images, indicating that there is still some information can be obtained from high-resolution images.

We too performed such an experiment for face recognition. While chapter 3 discusses the experiment and its setup exhaustively, the results are relevant here.

Train	Test on High-res	Test on Low-res
High	72.1	17.5
Low	23.6	48.3

Table 1.2 This experiment (detailed later in chapter in 3 used a very deep state-of-art CNN to recognise the identities of faces. Clearly, not only is the high-resolution domain very different from low-resolution domain, it is more conducive to learning discriminative features using a CNN.

While these numbers show that the two domains are indeed different, the results alone can't be used as evidence to argue that machine learning on the low-resolution domain is harder. It could just be that the popular methods are more tuned for the HR domain and other methods, perhaps just as simple, could exist for the LR domain. To show that it does indeed become hard to discriminate in the LR domain, we look at some cognitive science experiments which seem to suggest that even humans find it hard to work with low resolutions.

One study to refer to is [53] who studied covariates that influenced face recognition by humans. Among the covariates (like age, gender, expression of the person) considered was the resolution of the image. They conclude without doubt that increasing resolution increases recognition performance for both humans and machines. They found that while recognition was good at face resolutions between 32^2 and 64^2 pixels, it was very poor at resolutions below 16^2. They also suggested the existence of an "operational lower bound" on the resolution of the image below which recognition seemed to become much harder.

Another similar study was performed by [4] who varied the resolution of faces and studied performance by humans and machines for the task of face recognition. A plot of the accuracy vs face size, borrowed directly from the paper, is shown in figure 1.3. Now only does it show that there is a lot more error involved with the lower resolutions, there is a threshold resolution above which there is little improvement that adding resolution brings about. While it is no requirement that AI systems perform in the same way humans do, these results suggest that low resolution vision would be harder.

[79] performed a thorough study of the resolution at which humans can identify scenes, detect objects and localise people in scenes. Their study, which gave the plot in figure 1.2, noticed a dramatic increase in recognition correctness with resolution for small resolutions for many scene understanding tasks. An interesting point to note is that the performance on grayscale images is poorer than RGB images. The authors suggest that humans need approximately $32 \times 32 \times 3 = 3072$ dimensions of visual data (either grayscale or colour) for scene understanding tasks.

The other interesting observation is the presence of "critical resolution" above which high level tasks become easier. [12] observe a "bimodal behaviour" where the performance is stable for medium and high resolutions but drops for resolutions below a threshold. The community accepts [12, 79, 4, 53, 83] that this binary, the resolution below which we regard an image to be low resolution, is 32×32 pixels.

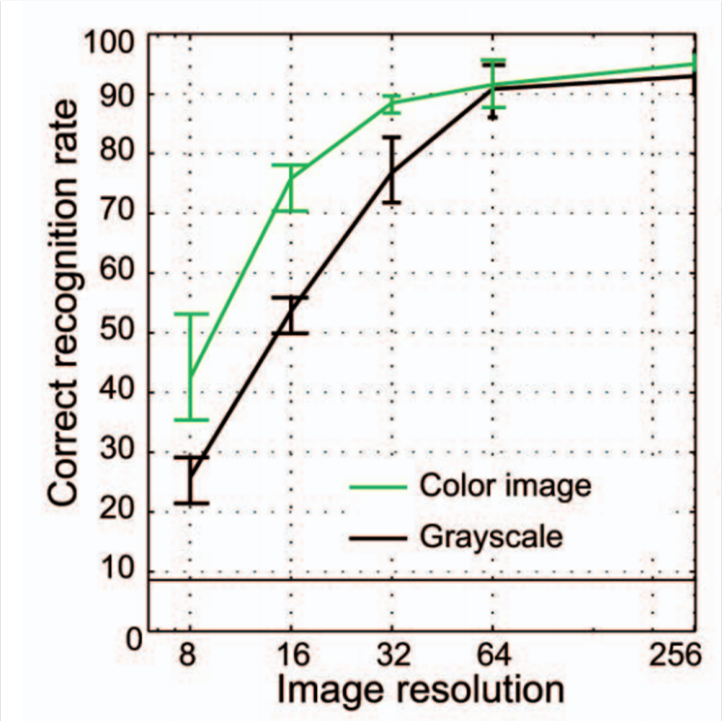


Figure 1.2 This plot from [79] shows the relationship between resolution, and effectively number of pixels of information, and scene recognition performance by humans. For colour images, there is only a 7% drop from images of 32×32 resolution to images of 256×256 resolution despite packing only 1.5% information present in the high-res image.

[79] ask the question of how big must an image dataset need to be for robust recognition and answer that it must be at least 32×32 sized. Observe in figure 1.2 that despite a 32×32 image being only 1/64 th the resolution of a full-res image, the drop in performance is just 7%. [79] also showed that while individual objects were still hard to recognise, when the entire scene was presented to humans, they could use scene and contextual information to localise objects and recognise people. They also noted that semantic instance segmentation was hard for humans at this resolution. In our experiments too, particularly in chapters 2 and 3, we use the 32^2 resolution threshold to guide our definitions of small objects and tiny faces.

Now that we have seen that even humans, despite there being evolutionary advantage for being able to recognise objects at very low resolutions, find scene understanding hard on tiny images, it is rational to expect that machine learning systems also struggle. On the other hand, convolutional neural networks have achieved tremendous successes on many visual tasks, even occasionally surpassing human-level accuracies. This subsection will deal with why CNNs as they are used today might not be able to replicate their successes on high resolution images in lower resolutions.

- One possibility for why deep learning methods don't work just as well for low resolution images is that the signal of low-res regions becomes too weak after successive pooling layers. Let us con-

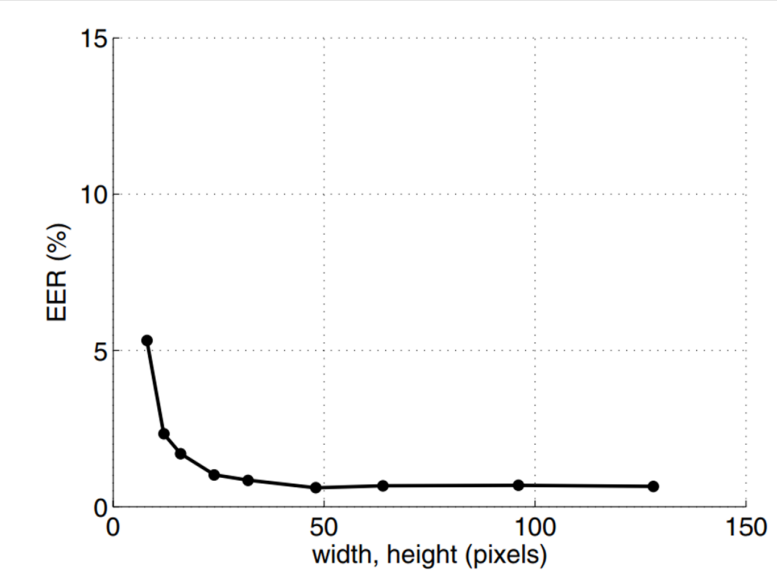


Figure 1.3 This graph from [4] shows face recognition accuracy by humans as a function of resolution. Notice that the error sharply drops for low resolutions and almost remains constant after a specific 'operational threshold' of 32×32 square pixels. In this thesis, any region that is smaller is considered to be of low resolution. [4] observe that performance of machines follow the same trend.

sider the very popular VGG16 network, one of the most widely-used CNN architectures, capable of obtaining features good for many vision tasks. It has five convolutional layer blocks with a max-pooling layer at the end of each block. The max pool operations are done with a kernel of size 2×2. This results in a decrease in the height and width of the input activation responses by a factor of 2. Therefore, if an object occupies 32×32 pixels in the original image, after five max-pools, its size is 1×1 (a drop by half after every max-pool). This 1×1 is the upper bound of the size of a feature assuming that 32×32 to be our upper bound on region-of-interest size. If the low-resolution region of interest is itself much smaller, it may not show up at all in the final activations. In encoder-decoder architectures for tasks like semantic-segmentation, image captioning, etc, the decoder will have to make do with just 1 pixel of information, which is in most cases, hardly sufficient. Later network architectures used large-strided convolutions instead of max-poolings to achieve greater depth. These suffer from resolution-loss as well. The reason for even using such layers is to increase the receptive field of future layers. They give deeper layers access to more global and contextual information which can be used for better disambiguation.

- This is related to the other problem of why CNNs don't work well with low resolution. Like human perception, CNNs look at objects in hierarchies. The lower layers act as corner and edge detectors and Gabor filters, while the latter layers learn relations and relative positions of parts of objects that form a complex object. When the object itself is in low-resolution, the network can not divide it into parts and hierarchies (because each part will be incredibly small). As an

illustrative example, hand recognition might look for presence of fingers and a palm in the mid-level and look for fingers sticking out of a palm in the high level. While it is easy to learn fingers and palm shapes, learning robust filters that capture all the variants of hand poses and gestures in just one level is difficult.

Now, knowing these problems, can we modify existing architectures to perform well on low resolution images? Clearly, the deep learning techniques in vogue have been designed keeping the high-resolution domain in mind. The Imagenet dataset has images of resolution 600 pixels wide on average and the size of an object in the PASCAL VOC dataset is 120 pixels. Here are some possible hyperparameters that can be tuned keeping this in mind:

- the filter sizes - if we want our filters to look at a larger part of the image, one way would be by using larger filter sizes. Also, we observe that in low resolution, the edges are thicker and the image itself looks blockier. Small 3x3 filters may not work for the lower layers because of this.

- number of filters - while it is true that the representation of a $32\times$ 32 object may be at most 1×1 in one activation map, we can use a lot more channels, and have a more flattened representation of the same information. Therefore, the feature representing the image will have size $1 \times 1 \times k$ where k is the number of channels.

- depth - it may be the case that the low-resolution space is densely entangled and several layers of non-linearity are required to learn good discriminative features. Hence, the CNN needs to be stacked with more layers and non-linearities.

In a disappointing development, [83] experimented with modifying all three of the above when doing a complete sweep of hyper-parameters for training a CNN on low-res images. Using larger filters or wider and deeper networks "brings little benefits" they conclude. The experiment they performed involved using an AlexNet-like architecture with three hidden layers but varying filter sizes and number of filters for image classification on the CIFAR-10 dataset. The errors on the LR domain were nearly double the errors on the HR domain for almost all choices of hyper-parameters.

One outcome of this experiment is that we got some intuition about what kind of features are learned by a deep CNN network trained on low-resolution images. [83] noticed that pixel intensity histograms were learned by CNNs for low resolutions. This outcome suggests considering classical histogram-based linear features for low-resolution images. However, all recent works that experimented with the most successful pre-deep learning methods reported superior performance with deep features. [12] used Fisher vectors over variants of SIFT features and found them to perform poorer than naive vanilla AlexNet features for low-resolution fine-grain classification problems. [9] compared the performance of deformable parts models to classify low resolution region proposals with AlexNet and VGG16 and found the latter two to be significantly more successful.

1.3 Related work

This section will look at some attempts in trying to use deep learning for low resolution images and begins by looking at some resolution-specific methods. Next we see methods that try to leech from the successes of deep learning methods on HR images to perform better on LR images. Procuring high resolution images is not hard given the explosion of visual content in the internet. Creating a low-resolution version of a high-resolution image is as easy as going through a trapdoor and only involves a downsampling interpolation operation. We look at approaches that try to learn the inverse problem: feature transfer and then even domain-transfer.

1.3.1 Resolution-specific networks

Multiscale Cascade CNNs introduced in [85] were useful for many localisation tasks. A multiscale cascade CNN splits the ranges of resolution into groups and a separate network handles each resolution. The suggested split of resolutions among four networks is 10-32, 32-120, 120-240, 240-480 pixels of width. Further, each range is split into three "scale ranges". For example, the 120-240 range is split into 120-160, 160-200 and 200-240. During training, each of the networks is trained with patches of corresponding to its dedicated resolution range. Because of successes of multitask learning and to send a stronger gradient, the network is also made to predict which of the three "scale ranges" the input patch belongs to. If training involves negative samples, the predicted "scale range" is -1. This approach helped [85] create a competitive benchmark for the WIDER face localisation dataset they introduced, and the approach they used inspired many other resolution-specific architectures like [27].

1.3.2 Staged training

Training resolution-specific networks does not use the presence of easily available high-resolution images. As observed in 1.1 and 1.2, there is a small amount of shared information between the two domains. One simple way to improve the performance on low resolution images using high-resolution images was suggested in [62]. The method requires high-resolution labeled data available during training, but it needn't be the higher-resolution counterparts of the low-resolution images. The method first is trained on the high-resolution auxiliary dataset. Then it is fine-tuned on a dataset of lower resolution (this can be a downsampled version of the high resolution dataset) with a higher learning rate. This model is then finally trained on the low resolution dataset. Such a approach was observed to perform better than mixed-training which constructs batches that have equal amounts of high-res and low-res images. Very recently, [75] suggested learning with privileged information where some side information like labeled high resolution images can guide the training of low resolution images. This sort of curriculum learning helped the model learn more discriminative features than just training on specific resolutions or mixed-training.

1.3.3 Shared feature representations

A more successful approach, used by [68, 10] and [83], maps features of both high and low resolution images to a common shared representation. They noted that it was very hard to bring both features to exactly the same feature space and hence only needed it to be approximate enough. A "tightly coupled" representation involves the same set of filters being learned to perform super-resolution of a low-resolution image and classification of a high-resolution image. A "partially coupled" modification has two networks for the above tasks sharing a majority of their filters. During test time, the filters part of the network that learns from input high resolution images is discarded.

1.3.4 Super-resolution

Since we know that our deep networks perform very well for the domain of high resolution images, this technique asks if we can map an image from the low dimension to the high dimension and then use the success of the high resolution domain. This domain transfer problem is not easy: we have to somehow add information and detail to upsample an image. Problems like these have a major challenge - there are multiple HR images that can be formed from a single LR image. Unless trained in a special fashion, neural networks will output the mean of the several possible HR images. This might be blurred and not contain sharp low level information like texture. But in recent times, the super-resolution algorithms have been remarkably accurate and provide sharp images. There are some works which use super-resolution to handle low-resolution. For instance, [15] showed that super-resolution improves the performance of low-medium resolution images over several visual tasks such as contour marking and semantic segmentation. [72] modifies Dynamic Filter Networks for classification-driven image enhancement. Very recently, [6] prepended a super-resolution network before a standard classification network and observed that this gave a fillip to classification. The super-resolution network is trained separately and fine-tuned during the classification step in the end-to-end network.

While most of these methods address only the problem of classification and synthetically created low-resolution images, it is we who first use super-resolution to improve generic scene understanding problems. Even if the image is of medium-high resolution, the region of interest could be of low resolution. We identify such scenarios and improve methods that ignored low-resolution. Our other contributions are described in the following section.

1.4 Summary

Thus far we have seen that there is little similarity between the LR and HR domains and that conventional vision problems are much harder in low resolution for both humans and machines. Many researches concur that there is a sharp cliff at around 32^2 pixels in the plot of recognition correctness versus resolution. We also saw that CNN-based deep learning methods get very mediocre accuracies on low resolution whilst being better than classical hand-tuned features. This is explained by the scarcity of

mid and high level information that can be extracted hierarchically by a CNN. Some ways to ameliorate this problem is by using resolution-specific networks or through super-resolution.

1.5 Contributions

We discuss the low-resolution and discuss why CNN-based methods perform poorly on low-resolution images. We identify important-yet-understudied problems in scene understanding that are adversely affected by low-resolution and through experiments, explain how to overcome them. More specifically,

- We theoretically examine the problem of anchor-box proposal generation in a region proposal network used in object detection. We discuss how to tune the anchor box sizes for different object sizes in general and show an example of how to pick anchor box sizes on a small-object dataset. We show that our choices of anchor box sizes for the small object dataset are not only adequate, but necessary. Through experiments we show a marked 4% improvement over the baseline region proposal network performances of faster RCNN and Chen et. al. [9].

- We propose a simple-yet-effective method to better classify proposal regions by super-resolving them. We demonstrate that this step adds valuable information to the image that enhances proposal classification and thereby better small object detection. We show improvement over our baselines of vanilla faster RCNN and an upscaling (but not upsampling) based method.

- We discuss the case of semi-structured scenes like faces in lecture-halls and social settings. We discuss the nuances that such scenes pose and why they are hard compared to other datasets.

- We study the problem of face detection when there are hundreds of faces of various scales, resolutions, illuminations, poses and expressions to empirically identify the best detector.

- We progressively infer that cross-training in a one-size-fits-all network produces very mediocre results and resolution-specific networks give the best results for semi-structured scenes.

- We propose a simple and novel method to improve the accuracy of modern fully convolutional networks for the task of semantic segmentation on road scenes by showing how to use unlabeled high-resolution images to improve semantic segmentation for street-scene datasets. The method has the potential to be made end-to-end trainable.

Chapter 2

Improving Small Object Detection

2.1 Introduction

Though the problem of object detection in natural scenes has seen a lot of research, especially since the development of deep ConvNets, it is far from being solved, particularly for the case of small objects. An object is considered small if it occupies only a tiny portion of the image (less than 1% of the image area). This problem is very relevant in many of the challenging research applications of today - like detecting pedestrians, traffic sings and cars on roads and areal imagery.

Detecting small objects is a challenging task because of the following reasons:

- Firstly, it is very hard to distinguish small objects from generic clutter in the background. This makes it hard for many of the standard detectors that rely on 'objectness' due to the drastic increase in the number of possible locations.
- Secondly, the activations of small objects become smaller with each pooling layer as an image passes through a standard CNN architecture like VGG16. For example, if an object has a size of 32×32, it will represent at most 1 pixel after the block5_pool layer in VGG16. Such activations can be easily missed.
- Thirdly, most small objects have simple shapes that are not decomposable into smaller parts. On the other hand, popular CNN-based detectors excel at learning hierarchical features.
- Lastly, there is no large publically-available dataset for small objects. While MS COCO[50] and VOC2012[19] have specific instances of objects being small, there are not any dedicated large datasets for small objects. Also, much of the prior experience and intuitions are on datasets with larger objects. While the mean Average Precision using the state-of-art end-to-end detectors on a dataset like PASCAL VOC is 76.3% [66], the state-of-art on a dataset with only small objects is just 27% [9].

A class of popular detection techniques in recent years involve suggesting several object proposal regions which are then classified by a deep CNN model [22]. These techniques are successful because

Figure 2.1 The problem of small object detection is hard because of a much larger search space, background clutter and a weak signal after passing through standard convolutional layers. For example, the mouse (in the green box) is a small object and is hard to spot among the various other objects of similar sizes present, even for humans.

the features obtained using a deep CNN are more discriminative than hand-engineered features. Unlike earlier times, when dense sliding windows were used to look at probable object regions, algorithms that look at low-level cues suggest much fewer and sparser windows. These proposal generation methods made way for a Region Proposal Network (RPN) [66], which was found to not only generate better proposals, but also greatly quicken the detection process when the weights of the convolutional layers are shared with that of the detector. In our work, we use a similar approach: an RPN generates proposals which are then classified by a deep CNN. The RPN as used in the de-facto standard detection algorithm, Faster RCNN [66], misses several small objects because of the large size of anchor boxes. Keeping this in mind, we study the size of anchor boxes for a dataset. We then show that this choice of anchor box size beats other existing methods.

Works like [17] suggest that the classification performance increases with the image size. One way to approach the problem would be to upsample the entire image and apply standard techniques on this image. However, the computational cost increases exponentially as the size of the image increases. Instead, we can upsample small proposal regions. Here, we take inspiration from recent works that convert low resolution images to high resolution by hallucinating the intermediate values. Though we are introducing an intermediate harder problem, even incorrect hallucinations might help in better feature-learning by ensuring there is enough high-frequency content in the image. We develop a method that

upsamples proposal regions with the hope of improving the overall classification performance.

The main contributions of this work are as follows:

- We formulate finding the appropriate sizes of the anchor boxes mathematically and perform detailed experiments to show the effectiveness in their choice. We show that this gives us the state of the art end-to-end trainable network for this dataset.
- We show how network-based super resolution techniques helps better classification of small regions-of-interests.

2.2 Related Work

In the pre-deep learning era, works used specially-crafted features for problems like vehicle detection in aerial imagery [64]. However, since the emergence of deep learning, the task of learning discriminative features has been transferred to CNNs.

Many approaches have emerged in recent times that do not use region proposals. YOLO [65] divides the image into a grid and predicts class labels and bounding boxes for each cell of the grid. Another interesting approach is Single Shot Detection [51] which fixes boxes of various scales where objects may lie and scores presence of objects for each such box during test time. However, proposal-based methods have been shown to outperform all proposal-free methods as far as recall and accuracy are concerned [29]. [80] and [94] were popular methods for proposal generation that used low and mid level features. The idea of using deep networks to suggest proposals has gained traction in recent years. While Deepbox [43] reranks proposals generated by Edgeboxes [94], DeepProposal [21] uses an inverse cascade that goes from the final to the initial convolutional layers of the CNN. The Regional Proposal Network introduced in [66] can share convolutional layers with the classifier network. Here, anchor boxes of multiple scales and aspect ratios slide across the feature map from the last convolutional layer. The RPN acts as an attention mechanism and tells the detector where to look.

[22] was made fast in [23] with the introduction of the RoIPooling layer which maps images of any dimension to a feature map of fixed dimension. Faster RCNN [66] is essentially two components - an RPN which feeds to Fast RCNN. All these approaches predict a bounding box and probability of belonging to that class for every class. It was observed that sharing weights between the proposal network and detector not only significantly reduces running time, but also improves performance.

Several detectors [39] [24] [84] have emerged that build upon the faster RCNN framework. However, most of them only fleetingly mention the case of small objects. [18] [27] and [47] look at modifying the fast RCNN architecture for the problem of logo, face and pedestrian detection respectively, all having instances of small objects.

Small object RCNN [9] is perhaps the first paper to focus on the problem of small object detection. They introduce a small dataset, an evaluation metric and provide a baseline score. They suggest minor

modifications to the Region Proposal Network and show an improvement in recall and mean average precision. They go on to argue that the RoI pooling layer may not preserve much information of small objects and hence follow the RCNN framework. In this work, we build upon their ideas and show how to make changes to perform just as well in an end-to-end pipeline.

The problem of image denoising and image super-resolution are well studied and have seen numerous approaches in the deep learning era. [16] showed how to train an end-to-end neural network for the task of Single Image Super Resolution. [73] and [37] use recursive CNNs and sub-pixel level convolution while [54] uses an auto-encoder model. We use [54] because it's relatively easier to experiment with it.

2.2.1 Overview of Faster RCNN

Faster RCNN follows the same pipeline as RCNN and Fast-RCNN, namely first generating a lot of proposals and classifying these proposals. But unlike the other two, Faster RCNN is completely end-to-end trainable which makes it perform much better, especially with proposal generation. It has the other advantage that the proposals need not be saved on disk. While the entire faster-RCNN networks fits on one GPU, its slower counterparts require proposals to be stored on disk. Because of the sheer number of proposals, for a 5GB dataset like Pascal VOC, they occupy about a 100GB to store proposals. Faster RCNN is also as its name suggests, very fast and offers near real-time detection. The speedup is offered by faster RCNN being a single unified network and because an image passes through convolutional layers only once. Here, we will refresh two intuitions of this approach - the region proposal network and the RoI pooling layer.

An image is passed through a stack of convolutional layers to obtain a feature map. This feature map is used twice - once by the region proposal network and the other by the latter classification network. The region proposal network slides 'anchor boxes' over this feature map. Typical region proposal networks use many scales and aspect ratios for the anchor box. For instance, the default anchor box sizes are $128 \times 128, 256 \times 256, 512 \times 512$ where the aspect ratio is 1:1. More anchor boxes such that the area is same as these but aspect ratios now 1:2 and 2:1 are also chosen. These anchor boxes slide across the feature map and the region enclosed constitute a region proposal. The region proposal network now classifies each anchor whether it belongs to a background class or foreground class based on 'objectness' of the region. It also refines the exact bounding box over a potential object by regressing four parameters that define a tighter bounding box over the object. During training, if a proposal does not have any object in it, the regression loss is omitted. A proposal is said to have or not have an object if the Intersection over Union area with any groundtruth box is greater than a threshold.

The classification of regions is done by dense layers which predict two outputs - a score for each class and a background, and bounding boxes for each class. However, these networks need a fixed size input, while a bunch of convolutions on different-sized images produces features of varying length. To mitigate this, the classifier has a region-of-interest pooling layer which splits an input feature map into a fixed number of parts and applies max-pooling to each part. The length of the output of the RoI pooling layer is fixed by the pre-defined number of parts.

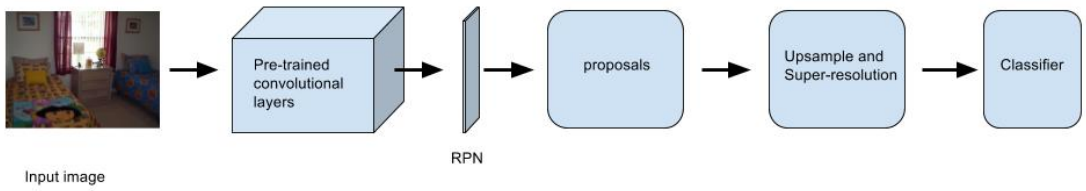


Figure 2.2 The pipeline for our approach - discriminative features that can be used for proposal generation and classification are obtained by passing the image through a standard pre-trained deep convolutional net. A region proposal network generates regions of interest based on the objectness of the region. These proposals are upsampled using a super-resolution network after which they are classified.

2.3 Approach

2.3.1 Proposal Generation

We make several modifications to the Faster RCNN [66] Region Proposal Network so that it performs well for the specific tasks of small objects. [23] suggested using powers of two like $128^2, 256^2, 512^2$ for anchor box sizes. While these anchor box sizes were shown to work for large objects, these are too large for small objects.

Like in [18], we theoretically estimate the size of the anchor boxes. The performance metric commonly used in detection to decide if a proposal is correct is to see if the Intersection over Union is greater than a threshold (typically 0.5). More formally, in Figure 2.3, if S_{gt} is the side length of the ground truth object and S_A is the side length of an anchor object, and d is the displacement of the two boxes, the IoU is defined as

$$\frac{(S_{gt} - d)^2}{S_{gt}^2 + S_A^2 - (S_{gt} - d)^2} \tag{1}$$

We want the quantity described in equation 1 to always be greater than a threshold t. In other words, we want $\min \text{IoU} \geq t$. We will only consider the case when $S_A \geq S_{gt}$. Solving for S_A, we get,

$$(S_{gt} - d)^2(1 + t^{-1}) - S_{gt}^2 \geq S_A^2 \tag{2}$$

The size of S_{gt} is dependent on our dataset. The worst possible overlap happens when the stride is largest. The value of d is dependent on the number of downsampling layers the image undergoes. Since we are fixing the anchor boxes after the fifth convolutional block, $d = 16$.

Thus, from equation 2, we get an upper limit for the size of an anchor box for a given ground truth image. This also gives us a bound on the size of the ground truth image for which our method will

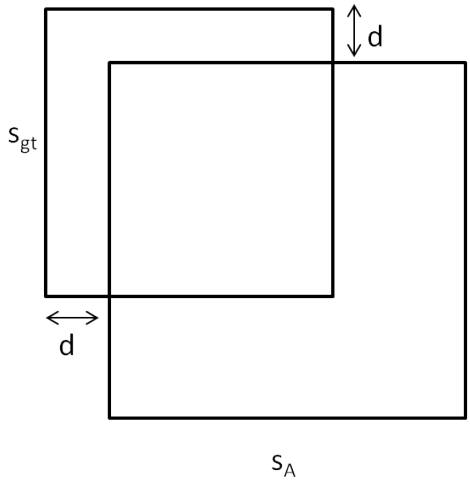


Figure 2.3 The choice of the size of anchor boxes is such that for all possible ground truth sizes and strides, an anchor box will have an overlap greater than a threshold.

work, since S_A^2 needs to be positive. We also note from [18] that $S_{gt}/S_A \leq 1/\sqrt{2}$. Here, we assume the bounding boxes to be squares. The same kind of relationship holds for other aspect ratios also. We analyze the size of objects in our dataset (Figure []4) and get our anchor boxes as $\{16, 25, 32, 45, 64, 90\}$.

Unlike [9], we train the RPN in an end-to-end manner to predict bounding boxes as well as class scores. This has been shown to improve performance [22]. Also, unlike them, we add batch normalization layers after every block in our VGG16 convnet.

2.3.2 Upsampling

Chen et.al. [9] argue that an RoI pooling layer loses discriminative features. They find that classifying up-sampled proposals gives a better performance than classifying patches despite the fact that aggressively upsampling adds undesirable artifacts and results in a noisy image. We instead leverage recent work in super-resolution to denoise the image.

A fairly common approach in super-resolution is to first upscale the low resolution image and use a CNN to denoise the image. The upscaling operation uses a hand-crafted filter like bilinear or bicubic interpolation. Such filters are but special cases of a deconvolutional layer [52]. A deconvolutional layer can be thought of as a convolutional layer with fractional stride. A network with deconvolutional layers might learn more complex, non-linear upsampling, specific to the dataset.

We use an implementation of [54] which uses a convolutional-deconvolutional network with skip connections. The usage of skip connections in the autoencoder makes the network easier to train while also ensuring that the deconvolutional layers can use the semantic information captured by the convolutional layers. The network is trained on a large dataset like Imagenet.We train a CNN on the upsampled

train images like in [9]. We use the VGG16 weights for the convolutional layers of a classifier which we use to classify the upsampled test proposals.

2.4 Experiments and Discussion

2.4.1 Dataset and Performance Metric

The dataset used in our experiments is the Small Object Dataset introduced in [9]. This is a collection of 4925 images from Microsoft COCO and the SUN dataset. Ten categories were chosen such that a typical instance of the object was no larger than 30cm in the physical world. Among all images which contained these classes, only those images which contained objects occupying a small area were chosen. The dataset is quite challenging for the following reasons :

- A significant percentage of the instances occupy less than 16×16 pixels (Figure 2.4). This is on average 0.2% of the image area. In contrast, datasets like VOC have objects that occupy 14% of the image area on average.
- There is class imbalance - while the category mouse has 1739 instances, the category tissue box has only 100 instances.
- The absence of high-resolution images for these categories is a major drawback. If high-resolution images were present, super-resolution network can be finetuned to a very high accuracy on these categories.
- The small size of the dataset with just 6000 train instances limits proper fine-tuning of existing methods for the case of small objects, let alone train full end-to-end systems from scratch.

We evaluate with the commonly used performance metric in detection: a predicted bounding box is considered a correct detection if the Intersection over Union (IoU) overlap with the ground truth bounding box is greater than 0.5. The performance of the whole detection algorithm is measured using mean Average Precision (mAP), which essentially denotes the area under the precision-recall curve.

2.4.2 Proposal Generation

We evaluate the choice of anchor box sizes against the default choice in Faster RCNN and those used by [9]. We use the standard faster RCNN framework with VGG16 as the backbone. The anchor boxes are attached to the conv5 layer of VGG16. The aspect ratios are the same as that of faster-RCNN, namely, 1:1, 1:2 and 2:1. The network is finetuned with a learning rate of 0.001 and a gamma of 0.1 for 50000 iterations. We compare the mAP upon taking the top 1000 proposals ranked on confidence of the proposal belonging to a non-background class for every test image.

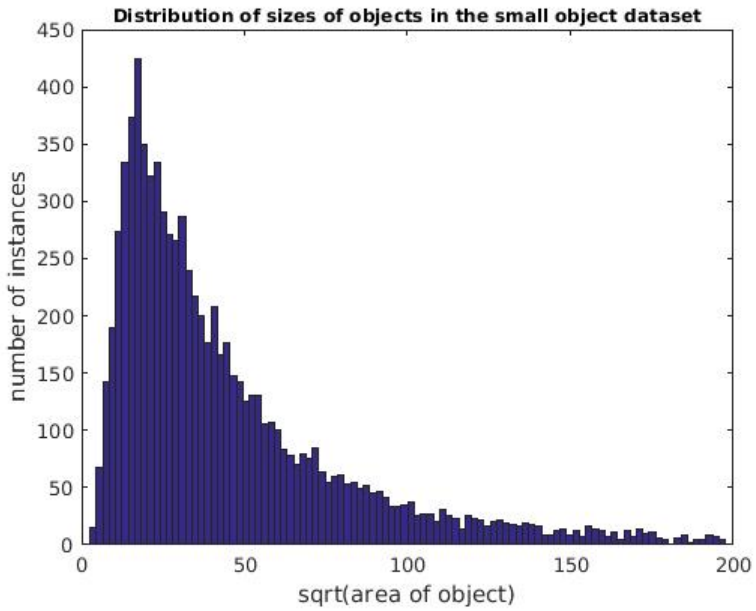


Figure 2.4 Size distribution in the dataset. Most objects are too small to be detected by the default anchor box sizes

While [9] uses the RCNN framework, we prefer experimenting with faster RCNN. The advantage of using Faster-RCNN is that apart from being much faster during testing and training, it does not require the storage of generated proposals which takes up a lot of memory. Also, we empirically found that attaching the anchor boxes to conv5 performs better than attaching to conv4 or conv3.

We see that our choice of anchors performs better than the default faster RCNN (table 2.2). This is expected since the smallest anchor box of size 128 is much bigger than all instances in the dataset. [9] choose anchor boxes of size 16,40 and 100. This performs much better than the default values. The anchor box sizes we propose cover the entire range of the small object sizes in the dataset.

To show that these anchors are adequate, we add two more anchor boxes of sizes 40 and 100. We observe that despite adding more anchor boxes, the performance slightly reduces to 21.9%. This is due to the larger number of proposals generated, which would include more proposals of generic objects that are part of the background. Because of the relatively simpler shapes of small objects, objects of these background classes might be confused for classes in our dataset.

2.4.3 Number of Proposals

We next compare the quality of the generated proposals with the number of proposals we consider for every test image and the choices for anchor box sizes. We continue to use the faster RCNN framework. Proposals that belong to a non-background class are ranked based on the classifier score and the top-k are chosen for the calculation of mAP.

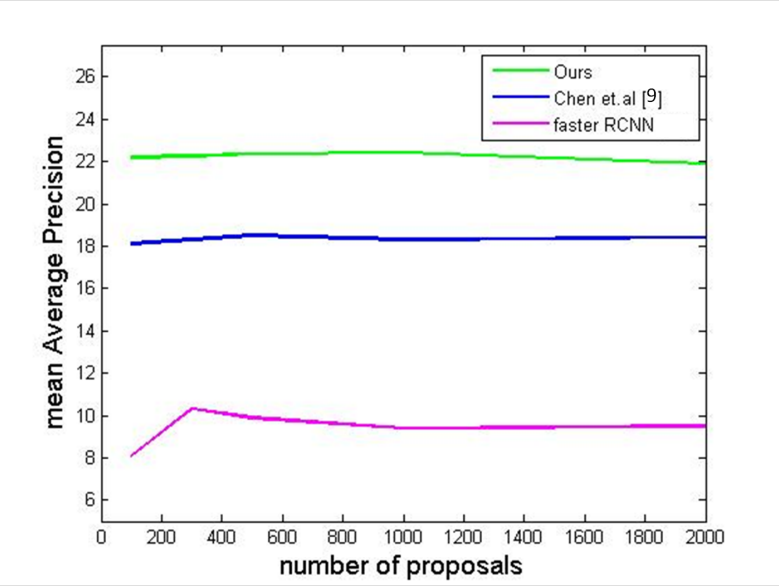


Figure 2.5 mAP vs number of proposals: our choice of anchors performs better than other methods even for much fewer proposals. It is interesting that the performance stagnates or even decreases with more proposals being considered.

Anchor Box sizes	mean Average Precision (%)
faster RCNN [66]	9.4
Chen et al. [9]	18.3
Ours	22.4

Table 2.1 Size of anchors vs performance. Our choice of anchors performs better than the default faster RCNN anchors and those used in [9] in the end-to-end faster RCNN pipeline.

As observed in Figure 2.5, we notice that our choice of anchor boxes beats [9] and the default Faster RCNN at all choices of number of proposals. We attribute the decrease in performance with more proposals to the observation that most true positives occur in the top few proposals itself, while there is an explosion in the number of false positives as more proposals are considered. These false positives arise because of the similarity in shape of generic background objects with the classes of interest.

2.4.4 Upsampling

To investigate the effect of using supervised up-sampling techniques, we parallel the approach followed by [9]. Here, we use the RCNN framework wherein we use our trained RPN with our choice of anchor box sizes to generate region proposals. We then upsample the proposals and use our trained classifier to rerank the scores for each proposal. The results are summarized in Table 2.2, where we observe that a super-resolution network improves performance. This can be attributed to the filters of standard

Method	Mouse	Phone	Switch	Outlet	Clock	T. paper
Faster RCNN	57.7	14.3	15.4	22.1	26.0	31.7
RPN + upscaling	56.8	16.4	31.1	29.4	31.9	29.4
RPN and super-resolution	60.1	16.9	16.2	23.5	30.3	34.1

Method	T. box	Faucet	Plate	Jar	Average
Faster RCNN	8.1	35.1	11.9	3.1	22.6
RPN + upscaling	23.4	31.3	9.3	4.2	24.8
RPN and super-resolution	12.8	38.0	15.2	4.7	25.2

Table 2.2 Results of our end-to-end method on the 10-class Small Object Dataset. The last column is the weighted average precision. The first row is the end-to-end trained Faster RCNN network with our anchors. The second row gives the performance with the RCNN pipeline and upscaling [9]. The third row shows the improvement with super-resolution.

convolutional networks which are typically trained for high resolution images, being less effective for low-resolution and blurry images.

2.5 Summary

In this chapter, we explored how faster RCNN as it stands can not handle the detection of small objects. One problem is that because of the vast difference in sizes of anchor boxes and the size of the object, the intersection over union of the two will only rarely be above a threshold. Here, we estimate the range of sizes of anchor boxes based on this threshold to show a marked improvement in proposal generation. We also notice that because of the low resolution of the small proposals, proposal classification is poor. To overcome this, we showed the effectiveness of super-resolving the proposals.

In this problem, low resolution manifested itself in the form of tiny objects in the image. To improve the performance of a CNN for classifying tightly bounded regions over these tiny objects, we used super-resolution with the hope of tackling some of the ill effects of low resolution. Faces in an image can also be seen as objects. In the next chapter, we discuss the challenges of face detection and recognition of low-resolution faces.

Figure 2.6 Exemplar results of our method on the small object dataset. The detections are shown in green boxes. The last row shows failure cases. In the first image, an armchair handle is classified as a mouse due to their similarity in shape. The second image shows a missed detection of the clock because it was too faint. The phone in the third image has two components quite far apart for our proposal generation method to consider as a single object.

Chapter 3

Detection and Recognition of Faces in Semi-structured Settings

3.1 Introduction

With recent advances in deep learning, facial recognition as a biometric is gaining traction over other methods. Even the latest version of the iPhone prefers to unlock the phone based on the face rather than the previously used fingerprint scanners. While facial recognition can be used to accurately identify several faces in a photograph taken non-obtrusively from an ordinary camera, contact-based methods like fingerprints or biometrics that rely on the iris or retina will require recognition to happen serially. This acceptance of facial recognition technology came about due to high accuracies achieved by deep-learning based works like [78, 70, 86] on popular datasets.

However, these models are only as robust as the datasets they are trained on. The large, in-the-wild recognition datasets of today do not capture the difficulties and nuances that "social" settings present. We look at semi-structured social settings as scenarios where the number of faces to detect and recognise is large. They are semi-structured because though the faces are in-the-wild and there is no explicit constraint on expression or pose, there is some structure- most of the people will be engaged in the same activity and at least a part of their face will be visible in a photograph. This has usecases in several applications like surveillance, tagging for social media, crowd-analysis, etc.
Social settings have their own unique challenges:

- Firstly, the number of faces in the image is much higher. For instance, while the popular detection benchmark dataset LFW [28] has about 2.5 faces per image, an image of people in a lecture hall can have 100 and upwards faces per image.
- Consequently there is a large variation in the sizes of the faces. In the lecture hall example, if a photograph is taken from near the stage, faces in the front row will be quite large. However, faces at the rear will be tiny.
- There is expected to be a significant difference in illumination because people would spatially spread out in a social setting and lighting wouldn't be uniform throughout. This is in direct contrast to datasets like LFW [28] and CFP[71] where the subjects are well-lit.

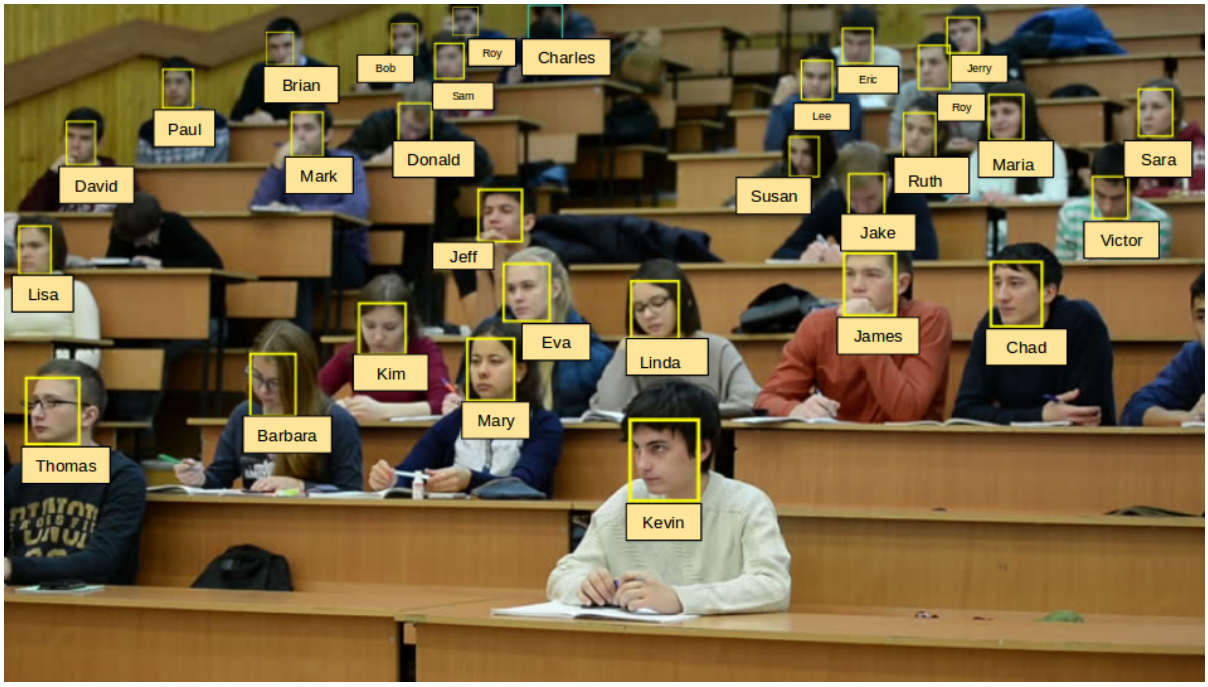


Figure 3.1 Lecture Halls are typical social settings. Notice the large variations in scale, expression and illumination of the faces apart from the presence of a large number of faces. Our goal is to detect and recognise all the faces.

- Popular datasets used by the community consist mostly of web-crawled images. The faces in such images are typically free from occlusion. However, in social scenes, occlusions and stark variations in expressions and pose are expected.

One setting that captures the challenges of number, scale and variations in pose, illumination and expression are lecture halls (like in Figure 3.1). Typical lecture halls have dozens of students who have very diverse facial attributes, expressions, facial orientation and pose. Because of camera limitations, only one part of the room is under focus in the photo. Also, there are changes in the faces of students with time, especially with hairstyles, accessories and ornaments, which make recognition harder.

One direct application of detection and recognition in education is automated attendance. This also enables other systems that build on top of recognition, like gauging listeners' receptiveness and engagement [36].

3.2 Related Work

While there are several recent works that deal with face detection and recognition for in-the-wild images, very few apply these to the scenario of unconstrained gatherings. For example, while works like [63] and [38] perform detection and recognition, sometimes in an unified pipeline, they are evaluated on datasets that don't reflect the semi-structured social settings well. In our pipeline, we have two steps:

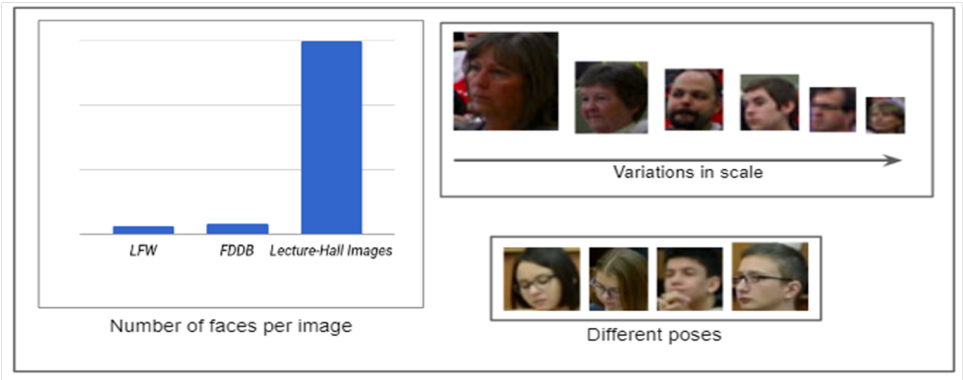


Figure 3.2 The sheer number of faces and the differences in scale, pose, illumination and occlusions make semi-structured social and lecture hall settings a hard problem

detection followed by recognition. The reason for a multi-step approach is that a human can fix the errors that crop up after every step so that they don't accumulate.

Face detection: Ever since the seminal work of Viola and Jones [81] for fast in-the-wild face detection, several works have upped the detection performance. Traditionally, methods that used cascading and deformable parts were used for detection. One survey of the most successful of these methods is in [85]. Most present-day methods use neural networks to generate regions of interest in the image. These are then classified based on 'faceness' [86]. While [49] and [35] were deep learning systems that achieved reasonable success, it is Tiny Faces [27] and Face Detection with Faster RCNN [33], two recent methods that have achieved state-of-the-art on common detection benchmarks.

Face recognition: Though there were early works like [13, 31] that used only neural networks for recognition, the idea of end-to-end neural networks for recognition caught on only after the introduction of large datasets. DeepFace [78], which trained a CNN on a large number of faces and identities, was one of the first works to assert the supremacy of CNN-learned features over hand-tuned features by achieving 97% accuracy on the LFW dataset compared to the earlier state-of-the-art of 70% accuracy. Some papers like [91] and [42] suggested face alignment based on key facial landmarks before passing through the network. However, later works like [70] argued that the networks were more robust without this alignment step, given a sufficiently large dataset. Some recent works used Siamese loss [77] and triplet loss [69] to learn a good representation that brought faces of the same identity closer which could be used for tasks such as verification. With the availability of a dataset the scale of CASIA [88], with over 10,500 identities and 494,000 images, simpler networks could be trained from scratch for the specific task of recognition. To handle the problem of scale, [45] and [46] proposed hallucination-based techniques to get a high-resolution mapping of the upscaled small face. While hallucination-based super-resolution techniques work for generic objects as demonstrated in the earlier chapter, identity-preserving super-resolution is much trickier, since even the state-of-art method by [90] achieves very

mediocre accuracies, only about 20% on standard datasets. Recent works such as [27] and [85] have achieved higher performance on face tasks by training disjoint networks for low and high resolution.

3.3 Scope

We want to explore a new class of scenarios: semi-structured social settings, where there are a lot of faces in the image, with a good fraction of the total faces are oriented towards the camera at any given point of time. However, the photos are captured in a non-obtrusive way which would put no constraint on the poses and expressions of the faces. We argue that face detection and recognition in this domain is hard, not only because of the large variance in scale, illumination, etc among the faces, but also because of lack of sufficient training samples in the datasets that are currently in vogue.

For example, detection algorithms are benchmarked on datasets that do not have the challenges of semi-structured social settings. FDDB [32], a commonly used benchmark dataset comprises mostly of celebrity faces without much variety in pose and illumination. Another popular dataset, LFW [28], has only 205 images, with most of face instances of medium scale. Moreover, both of these datasets offer less than 2.5 faces per image on average. The recent WIDER dataset [85] captures differences in pose, expression, illumination and scale has about 12.2 faces for every image and seems to be a more viable benchmark. However, most images in social settings contain many more faces per image. For example, we see from examples of lecture hall photos that the number of average face instances per image is at least twice that of WIDER. Also, we observe a significant variation in scale among the face instances in our dataset which the other datasets dont provide, with face width ranging from 30 pixels for the people at the back to 150 pixels for those at the front. This poses a great need to empirically find the best detector for this case.

We then investigate if simply training the state-of-the-art architectures for in-the-wild face recognition gives a good enough performance on semi-structured social settings. We find that the huge resolution variations prevent us from achieving performances similar to having used the same architecture for standard recognition datasets like LFW or CFP[71]. We then show the path for future study for this domain by suggesting improvements to the training process. We also discuss how to incorporate some post processing that helps enforce constraints like that a person is present at most once in a picture.

For such detection and recognition systems to find usage, the recognition step requires annotated instances of the identities. Here, we present a method for the easy annotation of large datasets so that the number of instances that need to be annotated is only proportional to the number of identities rather than being proportional to the total number of faces in the dataset.

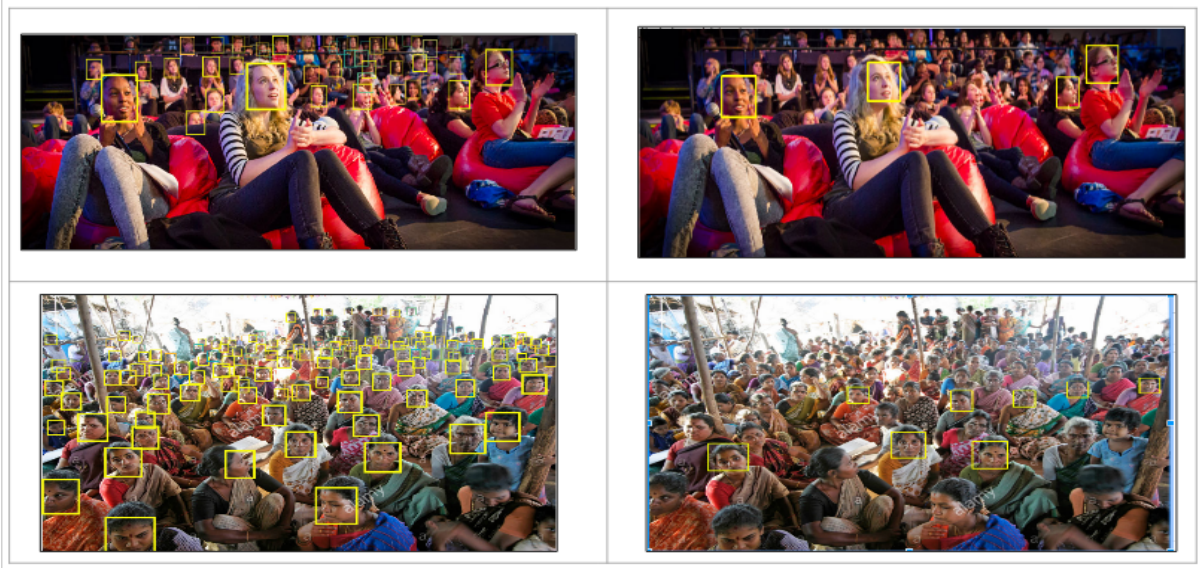


Figure 3.3 Exemplar detection results on a social setting. The results on the left are due to TinyFaces [27] while those on the right are due to Faster RCNN-Face [33].

3.4 Observations and Results

3.4.1 Face Detection

Significant strides have been made in the problem of face detection in the wild, more so since the advent of fast and accurate generic object detectors. However, none of these popular face detection algorithms was evaluated on challenging non-cooperative settings like those presented in a social setting. We therefore have to benchmark popular face detection algorithms on a dataset that is truly representative of social settings. For this, we gather images of the audience of lectures in a university. We use the Viola Jones detector that comes with OpenCV as a baseline for other detection methods. We use two popular algorithms: Faster RCNN face [33] and TinyFaces [27] as off-the-shelf detectors and evaluate them on various metrics. Faster RCNN Face finetunes the standard Faster RCNN on the WIDER train dataset. TinyFaces uses contextual and trains multiple subnetworks for different scale to achieve competitive performance on small scales. A bounding box prediction is considered as true detection if the Intersection over Union (IoU) with the ground truth bounding box is greater than 0.5, as commonly used in detection literature. It is considered a false detection otherwise. Using these local metrics, we compute the following global metrics as done in [38]:

$$\text{True Detect Rate (TDR)} = \frac{\sum_i^{1000} TD_i}{\sum_i^{1000} F_i}$$

where TD_i denotes the number of true detects in frame i and F_i is the total number of faces in the ground truth. This gives a measure of the number of faces detected for each frame.

$$\text{False Detect Rate (FDR)} = \frac{\sum_i^{1000} FD_i}{\sum_i^{1000} F_i}$$

where FD_i denotes the number of false detects in frame i. This tells the number of false positives per frame.

We use these to plot an ROC curve to compare the performance of the two detection methods 3.3. We find that while Voila Jones and Faster RCNN give recalls around 65%, TinyFaces captures 90 - 93 % of the faces. Specifically, it does not miss faces of small scale. Observing that TinyFaces significantly outperforms the other detection methods on our benchmark dataset procured locally, we proceed with using it to ease the task of annotating each face with the identity of the person.

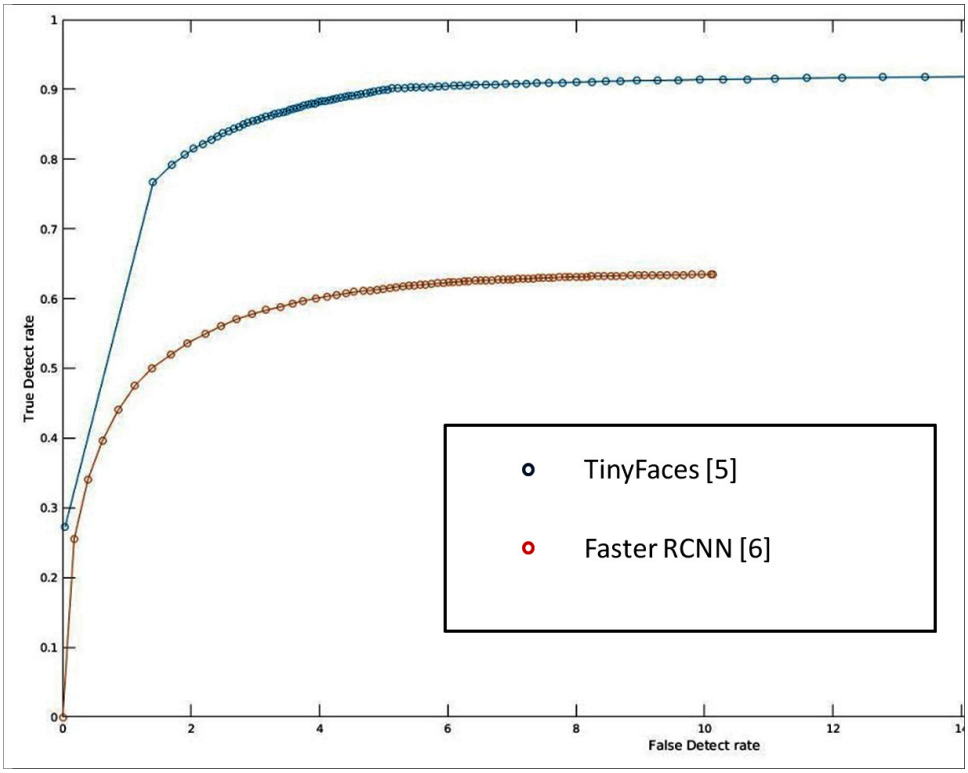


Figure 3.4 The ROC curve depicts the TDR vs FDR for the two best face detection algorithms on other datasets. We see that TinyFaces[27] works best in our case.

3.4.2 Face Annotation

To reduce the drudgery involved in manually assigning an identity to every face instance, we bring down the number of faces that need to be manually annotated from O(total number of face instances) to O(number of identities) using automation techniques. We create a web portal that allows for easy annotation of face images with identity labels. For each face, the system retrieves detected faces that are most similar with respect to identity.

The similarity between two faces is measured by the L2 distance between their respective VGG Face descriptors [59]. The feature representation produced by VGG Face preserves identity information since it is trained using a triplet loss that ensures that faces of the same person are closer to each other than faces of others. All faces with a similarity greater than a threshold are retrieved.

From the retrieved faces, the annotater selects a subset of correct retrievals. The identity label is propagated to these marked retrievals. An expert performs the much easier task of verifying the labels after the annotation step to ensure that it is error-free.

3.4.3 Face Recognition

After detecting faces, our system has to assign a label to each detection. Our approach involves obtaining a score for each identity per detection. LRFNet [26] has achieved the state-of-art performance on face recognition for all resolutions in recent times. We are following their approach and train an LRFNet-like architecture [26] for facial recognition as a classification problem. We observe that such an approach will not work well in social setting images. For example, we found that when the network is trained on faces from a lecture hall with a dozen images per identity, it achieves a very mediocre validation accuracy of 42%. However, when the same network is trained on an equivalent number of identities from the LFW dataset, even with much fewer faces per identity, the network reaches a rather high 91% recognition accuracy on the validation dataset. This clearly shows that the reason for the poor performance is the quality of the data we are using rather than quantity.

Type	Tested on	Trained on	Accuracy
Combined	Low+High Res	Low+High Res	42.75%
Cross	LowRes	HighRes	23.46%
Cross	HighRes	LowRes	17.53%
Res-specific	LowRes	LowRes	65.56%
Res-specific	HighRes	HighRes	72.14%

Table 3.1 Performance of a network trained and tested on the same identities, but different resolutions

We suspected scale to be a major factor that caused the method to fare poorly for the lecture hall setting. To verify this, we split the dataset into two smaller datasets: HighRes and LowRes (where image width is less than 32 pixels). When the same network is trained on images from this HighRes dataset, the validation accuracy jumps significantly from 42% to 72% as seen in 3.4.3. This evidences that the low resolution images are bringing down the performance.

We take inspiration from recent face-detection works like [27] and [85] to improve performance on low-resolution faces: these works train separate networks for different scales. We experiment finetuning on one resolution and testing on the other. The results of this experiment are in 3.4.3.

It is not surprising that there is a drastic reduction in performance when there is a difference in the resolutions of the train and test sets. Low-resolution images typically have blurrier and wider edges

and the filters learned by the networks for low and high resolution will be different. This experiment strongly points towards training different networks, from scratch, for different resolutions.

Also, there is still a significant gap of over 20% between the performances on LFW and the HighRes faces. This means that there are other factors that make social settings like lecture halls much harder. Face recognition works better with less variation in pose and expression. To understand the influence of pose and expression, we train the network on images from CFP [71], that has frontal and profile faces. The validation accuracies are shown in 3.5.

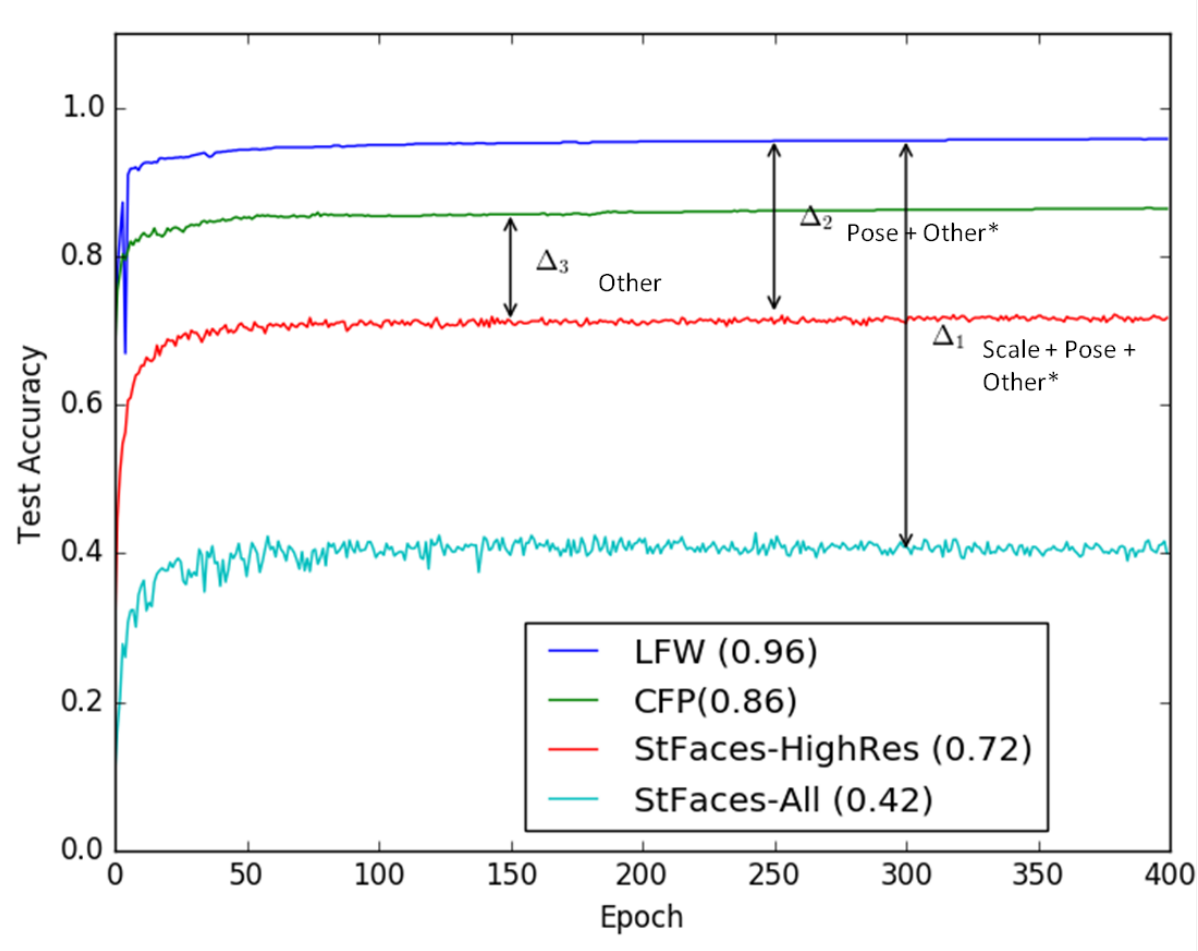


Figure 3.5 This Figure shows that there are other factors apart from scale and pose (like occlusion, illumination) that make recognition in social settings considerably more difficult.

This shows that actual social settings are significantly more difficult than those in benchmark datasets. We attribute the large difference to variations in scale, pose, expression, occlusion, illumination and camera quality.

As a post-processing step, we use the Hungarian algorithm to find the optimum matching based on the soft-max scores of each detection. This is because only one detection must correspond to particular

identity. This marginally improves our final accuracy from 70% to 72%. This is a marked improvement compared to the 42% achieved by a one-scale-fits-all network .

3.5 Summary

In our work we observe that the semi-structured case of faces in Social settings have not been explored in the past despite the widespread applications. This work is a step to understand this space. Neural-network-based methods are highly dependent on the datasets they are trained. Most datasets do not have the nuances and challenges that social settings have. Face detection is demonstrably hard because of the presence of tiny faces at a distance. The detector algorithm must be robust to handle stark variations in pose and illumination. We saw that low and high resolution faces are significantly different and a common, unified approach for classification will only give mediocre results. Semi-structured social settings have variations in occlusion, illumination and expression which makes recognition harder. There is also an inherent structure whose potential is yet to be tapped completely.

While the problem of face detection was closely related to object detection, a hallucination-based method that we used for low-resolution objects does not work well for low-resolution faces. Here we saw that scale-specific networks operate well on different resolutions of faces. There is a third kind of low-resolution that we encounter in computer vision - when we intentionally downsample an image for want of wider view, but we need low-level details later. This is the problem we explore in the next chapter.

Chapter 4

Better Semantic Segmentation through Super-Resolution

4.1 Introduction

One computer vision problem that has seen tremendous success in recent times is semantic image segmentation, where the goal is to assign a class label to every pixel in the input image. The task is important to the community because of its wide applications in autonomous driving and road-scene understanding for tasks like road boundary, obstacle and pedestrian localization, where we need exact shape and appearance.

Fully Convolutional Networks [52] revolutionized the problem with nearly every future state-of-the-art approach being an improvement over their idea. An FCN convolutionalizes the fully connected layers of standard image classification networks to produce a class-presence heat map at very low resolution. This activation map produced by the afore-described "encoder" network is upsampled to the same size and resolution as the input using a "decoder" network. This decoder network, made of deconvolution layers, learns non-linear filters for upsampling before feeding into a pixelwise loss. Skip connections between shallower layers that capture more local appearance information and deep layers that capture global and semantic information refines prediction to incorporate both "what" and "where" information. Approaches that succeeded this idea tried to find other ways to better capture local, high-frequency information while still getting contextual cues from a larger view of field of the image through the use of dilated convolutions and aggregation [89],[92] but essentially using the same encoder-decoder architecture.

A pooling layer lets the encoding CNN look at a larger part of the image, but at the loss of resolution, often to 1/8th or even 1/32th the size of the input image. Approaches thus far try to use dilated convolutions to lower the loss in resolution and super-resolve the produced activation map by trying to regain lost information by using coarse features from earlier layers. However, when objects are too small or thin, their presence can become negligible or even lost in the activations produced after multiple strided convolution or pooling layers. They can therefore be missed in the super-resolution process.

In this work, we propose performing a super-resolution not just before the decoding step, but before encoding as well. By giving an enlarged image, we mitigate the loss in information through a network,

while also allowing it to encode more local information. By focussing on a particular type of scenes (in our case, streets), we can avoid downsampling the image to very low resolutions when still give enough contextual information about the scene. The memory and computation overhead is overshadowed by the potential performance gain this method brings in and its simplicity.

With our method, it may be possible to advance the state-of-art in the road scene segmentation datasets like CamVid[5] and Cityscapes[14]. Our approach is modular and can be made end-to-end trainable. To train the super-resolution network, we utilize easily available high-resolution videos of streets, thereby leveraging high-resolution unlabeled data. We perform extensive experiments to demonstrate that prefixing a super-resolution network to a semantic segmentation network increases performance as the quality and scale of super-resolution increases. To summarize, our main contributions are

- We propose a simple and novel method to improve the accuracy of modern networks for the task of semantic segmentation on road scenes
- We show how to use unlabeled high-resolution images to improve semantic segmentation
- We argue that it is possible to achieve state of the art on popular benchmark datasets like Cityscapes by extending out method

4.2 Related Work

The problems of semantic segmentation and super-resolution have been studied extensively, but independent of each other. In this section, we trace research in super-resolution, semantic segmentation and using super-resolution to improve image tasks.

4.2.1 Super-resolution

The simplest attempts at super-resolution were by assigning the values of nearest neighbour pixel to the new pixels in the upscaled image. This was soon followed by biliear and bicubic interpolation which assigns a weighted average of the neighborhood to the new pixels. These simple interpolation techniques created several visual artifacts. They were particularly blocky and blurred. Methods like [74] and [87] attempted to negate these effects by using smoothness and gradient priors, or use global information by PCA [34], to restore the image. These approaches paved the way for the general trend that was to follow in image super-resolution: upscale the image first and then restore the high-frequency components by deblurring and sharpening.

Works emerged that leverage the availability of images of high-resolution images like [30] who matched patches with high resolution patches in a database. Manifold-based methods like [8] match patches from different resolutions by assuming that though low and high resolution images were from

different manifolds, the two manifolds had a similar local geometry. One of the earliest works that used neural networks for super-resolution was [93] who used a belief network to give a score of how much a low-resolution patch was similar to a high-resolution patch. [56] and [57] are survey papers that describe these earlier methods.

The first method that used the success of modern deep-learning was SRCNN [16], whose Super-Resolution Convolutional Neural Network was not just significantly simpler than the then-popular methods, but mapped a low-resolution image to a high-resolution image directly with one forward pass, thereby being several orders faster than multi-image approaches. Different deep architectures soon emerged like [52] which takes an interpolated image and negates the side-effects of upscaling with a multi-layer network. However, deconvolutional layers can, in principle, learn any linear upsampling [52]. Works like [55] used this to get end-to-end trainable architectures which gave their approaches better speed and accuracy.

These networks used a loss function that depended on the mean pixelwise error rather than perceptual quality. Though the approaches achieved a low peak-signal-to-noise ratio, output images lacked high-frequency texture information. Ledig et. al. introduced SRGAN [44] which introduces a perceptual loss and an adversarial training setup that produces high quality super-resolution and is currently the state-of-art and has been to shown to work for several kinds of domains like faces [3] and aerial data [7].

4.2.2 Semantic Segmentation

Before the deep learning era, semantic segmentation was done by using a classifier to assign a label to every superpixel of the image [20]. Contextual information was incorporated by [40] using CRFs and by [25] using structured prediction. With the advent of deep learning, methods used object detection to classify regions obtained by segmentation based on low and mid level features [22]. In [82], it was shown that it is possible to use recurrent convolutional neural network for pixel classification.

A big breakthrough came with the Fully Convolutional Network [52], which could provide dense class predictions for each each pixel. A convolutional layer replaces the fully-connected layers of a standard network trained for classification and is an example of how knowledge can be transferred from successful classification networks to other vision tasks. A decoder network is used to upsample the output back to the size of the original image. This operation can be seen as super-resolution: the encoder network produces a low-resolution feature map which is enlarged using deconvolutional or unpooling layers [1]. However, the high-resolution information lost in the downsampling is known to us during the super-resolution process unlike in the case of single-image super resolution. By using both long skip-connections from the encoding to the decoding layers, the upsampling layers recover the lost spatial information.

Many works have improved the performance of FCNs by introducing ways to capture a more global context. [11] and [89] suggested using dilated convolutions which allow convolutional layers to act on larger receptive fields without having to forgo resolution. PSPNet [92] uses aggregation of features at

multiple scales to provide both local and global information to the upsampling decoder network. GCN [61] employs long, thin filters so that the network performs well for not just dense localisation tasks, but also for classification.

However, these FCN-based models were computationally very heavy and could not fit in single GPUs and could not process frames in real time. To overcome this, some networks emerged that retained accuracy while being much lighter and faster. The improvisation of E-Net [60], namely ERF-Net [67] is the work we concentrate on in this work.

4.2.3 Usefulness of super-resolution in vision tasks

Super-resolution has been shown to improve performance in domain-specific tasks such as facial recognition [3], vehicle detection in aerial images [7], and object detection [48]. However, the objects in consideration are of low-resolution, unlike our algorithm which aims to map images that are already at high-resolution to a higher resolution. In [15], Dai et. al show how super-resolution can be used to improve performance in tasks such as edge detection, scene recognition and semantic segmentation. However, they only discuss using these tasks as an evaluation metric for super-resolution. Also, our approach can be made end-to-end trainable, unlike theirs.

4.3 Approach

Our approach can be seen as performing a super-resolution step before sending through a semantic segmentation network. Apart from allowing us to easily analyze the final performance by replacing either block with other off-the-shelf networks, this approach is end-to-end fine-tunable. We detail the various networks that we considered in the reminder of this section.

4.3.1 Semantic Segmentation

Fully Convolutional Networks [52] advanced the then state of art by over 20 % for semantic segmentation. In this network, the fully connected layers (like fc6 and fc7) of a VGG network are replaced by convolutional layers. While FCNs can accept input images of any reasonable size, the original paper had input images of size 500 x 500. At the end of the encoding step, the output has a resolution of 70 x 70, which is upsampled to 500 x 500. The upsampling is handled by a decoder network. In FCN, deconvolution layers, initialized with bilinear interpolation filters form the decoder network. A salient feature of FCN is using skip connections where features from block3 and block4 are fused with outputs of later layers. With three such skip connections, the three-stream FCN8s, because of the much lower stride, sees a marked improvement over FCN32s which has no skip connections.

In our approach, the input size to the network is much higher. For example, when testing it on CityScapes, images of which have size 1800×900 after super-resolution, we aim to use as much information contained in the image as possible. Hence, we use the full input images, but with a very

low batch size, so that the entire model fits in a GPU. Continuing with the same architecture, the output segmentation maps are bigger, but still smaller than our ground truth. Note that in our entire approach, we do not super-resolve the ground truth masks.

In our implementation, we finetune a network that is already proficient at segmentation at the original scale. We use the adam optimizer and trained for 150 epochs. For a majority of our experiments, we use ERF-Net [67], a lighter version of the same architecture to fit batches of larger images in a single 1080 Ti GPU. ERF-Net uses ResNets, which have much fewer parameters than VGG16 as its backbone. Another reason for their efficiency is the several linkages between blocks of the encoder and decoder networks. The decoder network also requires much fewer parameters with deconvolutional layers resizing input of size $H \times W \times C$ to size $2H \times 2W \times C/2$. Despite needing much fewer parameters, ERF-Nets achieve performances very near to state-of-art. We train with a really small batch size and the full resolution that we can leverage. For very high-resolutions, we crop the image into smaller chunks and and stitch them together. While this stitching can be improved with some post-processing, we do not use any in this work.

4.3.2 Super-resolution

Our hypothesis is that super-resolution as a preprocessing step improves semantic segmentation. To verify this, we experiment with a variety of super-resolution algorithms. Among the single image based methods, bilinear and bicubic interpolations are very popular due to their simplicity. Experiments suggest that even these methods might offer a small improvement. The problem with the interpolation techniques is that the kernels are non-learnable.

SRCNN [16] performs super-resolution by using an encoder-decoder architecture to denoise an image the is upscaled using an interpolation technique. Experiments in the original paper advocate more channels in the hidden layers. More specifically to such kind of super-resolution networks, the filter sizes are larger. In the simple SRCNN architecture, the three layers use convolution filters of sizes 9, 1 and 5. Later methods were variants of this network and mostly added more layers, skip connections [55] and feature aggregation to improve the performance.

What we shall henceforth refer as SRCNN is the Red30 convolutional autoencoder architecture with symmetric skip connections as described in [55], but also help in training the network faster and smoother flow of gradients to earlier layers. Such a network was only evaluated on smaller and artificially created low resolution images.

We also use a more recent approach that capitalized on two breakthroughs - ResNets and GANs, namely ResNet-SR described in SR-GAN [44], we use the ResNet SR pretrained using a GAN loss and finetune it using only the MSE loss for super-resolution.

To make our SuperSegmenter architecture end-to-end trainable, we simply append the semantic segmentation network to the super-resolution network. The SR network only denoises and removes interpolation artifacts such as blurring, ghosting and blockiness. Both the semantic segmentation and super

resolution networks are first finetuned for the higher resolution before the entire large network can be trained end-to-end.

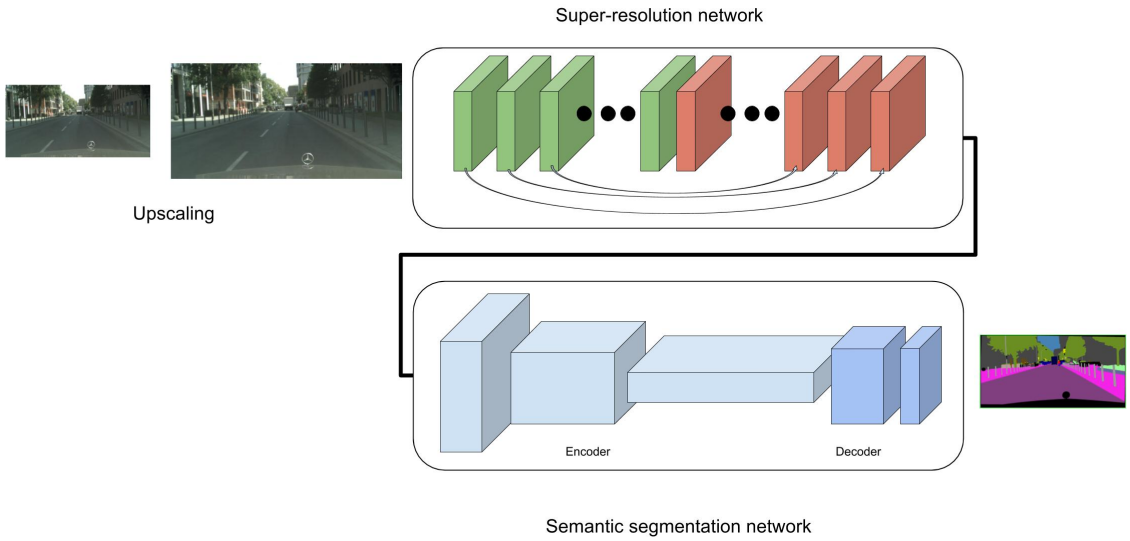


Figure 4.1 Our proposed architecture involved upscaling an image and then using a super-resolution network to deniose this image and then sending this through a fully convolutional network for semantic segmentation. The super-resolution and semantic segmentation blocks can be replaced by other networks that perform the same task, making the approach modular, simple to fine-tune and end-to-end trainable.

We use adam to train this 'SuperSegmenter' network. The SR networks are trained for 80 iterations and the segmentation networks for 150 iterations. We experiment by using ERF-Net for semantic segmentation, but we postulate that a performance boost will be observed in more recent architectures like PSP-Net as well.

4.4 Experiments

We perform experiments that investigate both the quality and quantity of super-resolution and quality of semantic segmentation to validate our approach.

4.4.1 Dataset and evaluation metric

The detaset we evaluate our metric on is Cityscapes [14], a popular benchmarks for semantic segmentation for urban roads. Cityscapes has 5000 finely-annotated images with 2975, 500, and 1525 images belonging to the training, validation and testing sets. It has object classes such as truck, car and sidewalk and stuff classes such as sky and terrain. The evaluation metrics used in this work is the mean of class-wise intersection over union.

For enhancing super-resolution, we use a dataset of 5 minutes of high-res videos collected in our urban neighbourhood.

4.4.2 Quality of super-resolution

We should suspect that the quality of super-resolution influences the performance boost brought in by our approach. This quality is typically measured by metrics such as Peak Signal-Noise Ratio, while these have been found to be insufficient in recent work [44]. We therefore compare our approach on frames upscaled by multiple super-resolution algorithms - bilinear interpolation, bicubic interpolation, SRCNN and SRGAN and notice a trend.

While bilinear and bicubic interpolation are "unsupervised" methods, SRCNN and SRGAN can leverage the availability of high-resolution images from the same domain. We perform the experiment on Cityscapes, super-resolving it on frames from a 5-minute video captured in our neighbourhood. The ERF-Net is once again trained for this high-resolution dataset. The results are in the table below. The baseline mean IoU without any upscaling is 0.69.

Interpolation	Bilinear	Bicubic	SRCNN	SRGAN
PSNR	34.14	37.08	38.04	39.91
Mean IoU	0.68	0.69	0.7	0.71

Table 4.1 Effect of super-resolution on Cityscapes dataset. This table shows that the observed trend holds for other benchmark datasets as well.

A higher PSNR implies that the signal to reconstruction error ratio is lesser, and hence, the super-resolution is of higher quality. Once again, we observe that higher the PSNR, higher the performance on both the metrics for semantic segmentation.

4.4.3 Upscaling factor

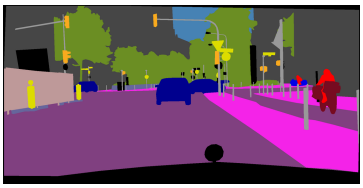
The next experiment looks at the accuracy of semantic segmentation as the size of the input image varies. The premise of this work is that as the size of a high-resolution input image increases, the amount of information available to the model increases and hence our model must perform tasks such as semantic segmentation better. To demonstrate this, we vary the input size to the semantic segmentation network from a half times downsampled to one and a half times upsampled and observe the effect on semantic segmentation.

The baseline is the performance on the standard input size without any resizing of the input image on ERF-Net. For the smaller scales, we resize both the ground truth masks and the input roadscene image using bicubic interpolation. For the higher scale, we upsample only the input image while the size of the ground truth mask remains the same. This upsampling is performed by super-resolving using SR-GAN, the best method as found in the earlier experiment.

original image

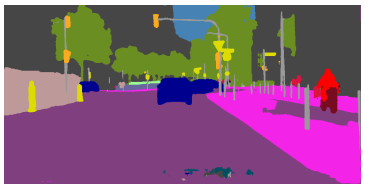
ISR through bicubic interpolation (PSNR 37)

ISR through SR-GAN (PSNR 39)



ground truth

FCN over bicubic interpolation, mIoU:0.69



FCN over SR-GAN output, mIoU:0.71

Figure 4.2 The above images show how the quality of super-resolution influences the semantic segmentation performance. The first row shows the input image super-resolved using bilinear interpolation and through the state-of-art SR-GAN. The corresponding images in the second row show the segmentation performances. Notice that the better segmentation is for the clearer and more accurate super-resolution, especially for the streetlights visible at the back.

The ERF-Net is then finetuned on the new scale of the training images before the scores on the validation dataset are calculated. We show results on Cityscapes, while using our collected road scenes to finetune super-resolution for this case. The results are summarised in the table below.

Scale	1/2 ×	1 ×	1.5 ×
Mean IoU	0.65	0.69	0.71

Table 4.2 Size versus semantic segmentation performance on the Cityscapes dataset. The trend where the increment provided increases with input image size is observed.

Our observations tally with our expectation of semantic segmentation performance increasing with the size of the image. From the above two experiments, we see that for high-level vision tasks such as semantic segmentation, fully convolutional networks work best on large, high-resolution images.

4.5 Qualitative Results and Discussion

The success of our method can be seen in figure 4.4 where we show semantic segmentation results with a standard ERF-Net and our SuperSegmenter network. Our method is slightly better in not just boundary prediction of foliage and buildings, but for being able to capture the thin streetlamps and traffic lights in the segmented output. Because of class imbalance and maxpooling, these are hardly visible at the depths of segmented network at the usual resolution. However, the information lost is compensated by the information gained when the input image is upsampled. It will be interesting to see

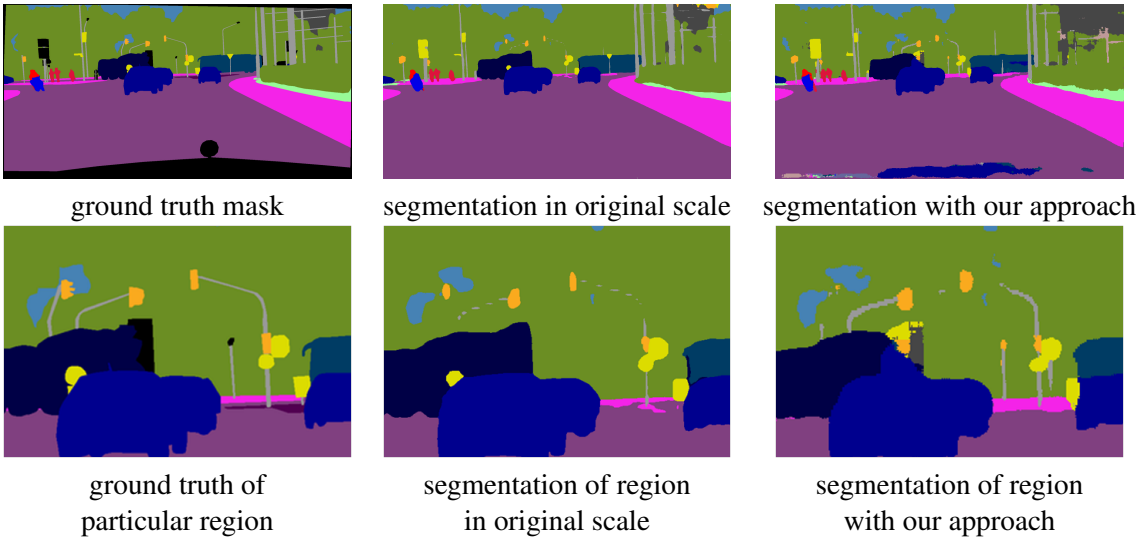


Figure 4.3 This figure shows how the segmentation mask produced by our approach (right) is closer to the ground truth (left) than the baseline (center). Specifically, our method performs better for thinner objects and is able to capture small variations in class boundaries.

if the approach will work in diverse settings where scene context have shown to aid in the dissolution of doubt between similar-looking objects but native to different environments.

4.6 Summary

In this chapter, we saw that super-resolving an input image before sending it through a semantic segmentation network improves performance. We attribute this increase to how signals decay with max-pool operations and the difficulty in regaining lost information. When focusing exclusively on street-view scenes, we are able to forgo the need for wide context. We take a popular ERF-Net architecture, prepend it with a autoencoder-based super-resolution network and show improvement on the cityscapes dataset. We justify the increased computational cost with an improvement in accuracy. We conjecture that the same approach can be used to improve the performance of even state-of-art architectures on road scenes.

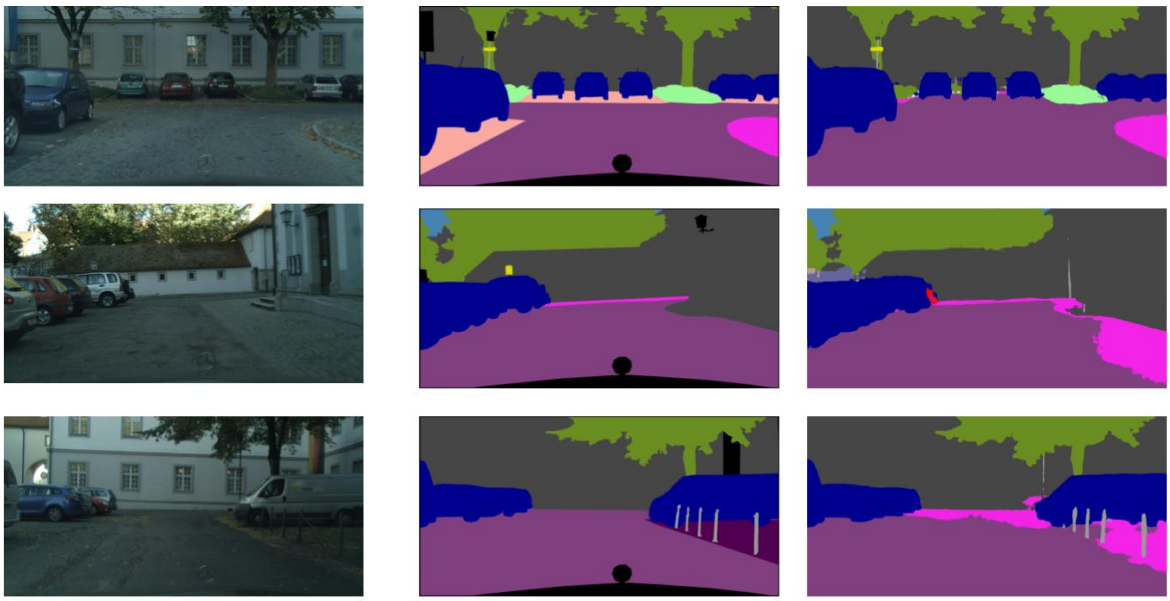


Figure 4.4 This figure gives some sample outputs for our algorithm. The first column is the input image, the column in the middle is the ground truth and the rightmost column is the obtained result. It is interesting to see that the network predicts a street pole towards the right for the second image even when there is no such pole in the ground truth. These are some of the side-effects of hallucinating using super-resolution. A similar tall, thin pole also appears in the third image, under the tree.

Chapter 5

Conclusions and Future Works

In this thesis, we identified that low-resolution held back higher accuracies in scene understanding. Learning discriminative features on the low-resolution domain is significantly more difficult than learning features for the high resolution, especially for modern deep convolutional neural networks. CNNs look at objects in hierarchies, are designed to trade resolution for a better field of view, and tuned keeping large datasets in mind. Hence, though they perform brilliantly for high-resolution images, they perform badly for low-resolution. Some ways to assuage this was to either map the low and high resolution to share features or transfer the domain of the source image from low-resolution to high-resolution.

While simple interpolation techniques create visual artifacts that adversely affect classification performance, super-resolving a lower resolution image gives the classification models a more familiar input. Not only does the network use easily available auxiliary high-resolution images to infuse new information, the super-resolution step hallucinates details, which despite being perhaps faulty, help Convolutional Neural Networks finetune learned discriminative features.

The popular object detection methods in vogue do not handle the case of small objects. Faster RCNN, which has beaten most other methods on a variety of datasets, also reels from this problem. Firstly, the region-proposal-network may be attached to too deep a layer where the signal of a small object might have died out. One way to handle this is to fix the region proposal network after intermediate blocks and choose the configuration that works best for the dataset in question. Secondly, the anchor boxes that the proposal network suggests are too large for small objects. Since the metric used depends on the area overlap, there are restrictions on the area of the anchor box, which we explore. Thirdly, each proposal region is tiny and needs to be super-resolved. With these methods, small object detection performance by a proposal-based method like Faster RCNN can be improved. We show that with these, we better the performance on a small object dataset.

There is still a lot of information in the image that is left unused to better small object detection. Cognitive experiments show that while humans can't recognize small objects when presented in exclusivity, when shown entire scenes, we could accurately localise and identify individual objects. This contextual information is crucial for better scene understanding. Many present-day algorithms do not allow increasing the receptive field for looking at a large part of the image without compromising resolution.

However, recent approaches in semantic segmentation that use spatial pooling and aggregate features from different scales and fields of view through dilated convolutions, when adapted to bounding box detection, might provide enough contextual information. Another approach to take this forward is to "super-resolve" features of the region proposal region directly rather than super-resolving images. This will cut down time from our method since this will require only one forward pass through computationally expensive convolutional layers compared to our suggested method that needs two forward-passes.

All the popular face detection and recognition algorithms were benchmarked on datasets that contained large, frontal faces that were well lit. Each image of these datasets had only about 3 faces per image. However, in real life, we come across several instances when we need to perform detection and recognition for hundreds of faces per image, and where there is some inherent structure in the organisation of people, like in lecture halls. In this thesis, we discuss and benchmark algorithms and datasets for detection. We also observe that because of illumination, pose and expression variations, and the inherent difficulty of learning features at low-resolution, recognition performance is hit. Through experiments, we argue why resolution-specific networks are the way forward.

Like in the case of object detection, context can help in better facial recognition in these scenes. There is a lot of unexploited structure in these loosely-structured settings. For example, when used for lecture halls, the system could learn that friends typically sit together and use confident recognitions of some people in a gang to disambiguate others. These relations and affinities between people are best represented by graphs. Like CNNs are invariant to translation and RNNs to temporal shifts, a deep network that processes this graph must be invariant to permutation. Another possible extension to this work is to gauge how interested students are in a lecture and identify tougher or relatively less-interesting concepts taught by an instructor.

When deep CNNs perform semantic segmentation of road scenes, they might lose high-resolution information about complex edges, thin objects and the like. One way to tackle this is by providing a higher resolution image as input itself. We observe empirically that the super-resolution step adds enough information to the image so that when the super-resolution network and fully convolutional network are trained in an end-to-end fashion, we get a performance gain. We show that the greater the resolution, the better the semantic segmentation, validating our claim. The disadvantage of the approach is that it requires greater GPU memory and more computational cost. Since both the super-resolution and semantic segmentation networks are both based on fully convolutional networks, it will be interesting to see if the two can be made to share layers and complement the other in parallel.

One source of low-resolution unexplored in this work is when the source camera is of low-resolution. Surveillance cameras might intentionally want to capture at low-resolution because of privacy concerns. A low-resolution video stream also has the advantage of needing much less disk space, and hence network bandwidth. Security cameras might also record at low frame-rates for more storage efficiency. While this thesis only deals with super-resolution along space, it will be interesting to consider super-resolving along time as well.

Related Publications

Improving Small Object Detection
Asian Conference for Pattern Recognition (ACPR), 2017 [Oral presentation, travel award winner]
Harish Krishna , C.V. Jawahar
CVIT, KCIS, International Institute of Information Technology, Hyderabad

Detection and Recognition of Faces in Semi-Structured Settings
Preprint
Samyak Dutta , **Harish Krishna** , Harshil Jain, Ashwin Sudhir, C.V. Jawahar
CVIT, KCIS, International Institute of Information Technology, Hyderabad

SuperSegmenter: Better Semantic Segmentation through Semantic Segmentation
Preprint
Harish Krishna , Girish Varma, C.V. Jawahar
CVIT, KCIS, International Institute of Information Technology, Hyderabad

Bibliography

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 35

[2] S. Baker and T. Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 5

[3] E. Bilgazyev, B. Efraty, S. K. Shah, and I. A. Kakadiaris. Improved face recognition using super-resolution. In *IJCB*, 2011. 35, 36

[4] B. Boom, G. Beumer, L. J. Spreeuwers, and R. N. Veldhuis. The effect of image resolution on the performance of a face recognition system. In *International Conference on Control, Automation, Robotics and Vision*, 2006. ix, 4, 5, 6, 8

[5] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2008. 34

[6] D. Cai, K. Chen, Y. Qian, and J.-K. Kämäräinen. Convolutional low-resolution fine-grained classification. *arXiv:1703.05393*, 2017. 11

[7] L. Cao, R. Ji, C. Wang, and J. Li. Towards domain adaptive vehicle detection in satellite image by supervised super-resolution transfer. In *AAAI*, 2016. 35, 36

[8] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In *CVPR*, 2004. 4, 34

[9] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao. R-cnn for small object detection. In *13th ACCV Proceedings*, 2017. xi, 9, 12, 13, 15, 18, 19, 20, 21, 22

[10] J. Chen, J. Wu, J. Konrad, and P. Ishwar. Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions. In *WACV*, 2017. 11

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 35

[12] M. Chevalier, N. Thome, M. Cord, J. Fournier, G. Henaff, and E. Dusch. Lr-cnn for fine-grained classification with varying resolution. In *ICIP*, 2015. 6, 9

[13] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 26

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 34, 38

[15] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. Is image super-resolution helpful for other vision tasks? In *WACV*, 2016. 11, 36

[16] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV 2014*. 5, 16, 35, 37

[17] C. Eggert, A. Winschel, D. Zecha, and R. Lienhart. Saliency-guided selective magnification for company logo detection. In *ICPR 2016*. 14

[18] C. Eggert, D. Zecha, S. Brehm, and R. Lienhart. Improving small object proposals for company logo detection. In *ICMR 2017*. 15, 17, 18

[19] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 2015. 13

[20] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009. 35

[21] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. Van Gool. Deepproposals: Hunting objects and actions by cascading deep convolutional layers. *IJCV*, 2017. 15

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR 2014*. 13, 15, 18, 35

[23] R. B. Girshick. Fast R-CNN. *arXiv*, 1504.08083, 2015. 15, 17

[24] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *arXiv*, 1703.06870, 2017. 15

[25] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 35

[26] C. Herrmann, D. Willersinn, and J. Beyerer. Low-resolution convolutional neural networks for video face recognition. In *AVSS*, 2016. 30

[27] P. Hu and D. Ramanan. Finding tiny faces. In *CVPR*, 2017. x, 10, 15, 26, 27, 28, 29, 30

[28] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 07-49, University of Massachusetts, Amherst, 2007. 24, 27

[29] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv*, 1611.10012, 2016. 15

[30] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 1991. 34

[31] R. Jafri and H. R. Arabnia. A survey of face recognition techniques. *Jips*, 2009. 26

[32] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, 2010. 27

[33] H. Jiang and E. G. Learned-Miller. Face detection with the faster R-CNN. *FG*, 2017. x, 26, 28

[34] C. V. Jiji, S. Chaudhuri, and P. Chatterjee. Single frame image super-resolution: should we process locally or globally? *Multidimensional Systems and Signal Processing*, 2007. 34

[35] I. A. Kalinovsky and V. G. Spitsyn. Compact convolutional neural network cascadefor face detection. In *CEUR Workshops*. , 2016. 26

[36] A. Kamath, A. Biswas, and V. Balasubramanian. A crowdsourced approach to student engagement recognition in e-learning environments. In *WACV*, 2016. 25

[37] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. *arXiv*, 1511.04491, 2015. 16

[38] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR*, 2015. 25, 28

[39] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: towards accurate region proposal generation and joint object detection. In *CVPR 2016*. 15

[40] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, 2011. 35

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2014. 5

[42] A. Kumar, R. Ranjan, V. Patel, and R. Chellappa. Face alignment by local deep descriptor regression. *arXiv:1601.07950*, 2016. 26

[43] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *ICCV 2015*. 15

[44] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 35, 37, 39

[45] J. Lezama, Q. Qiu, and G. Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *CVPR*, 2017. 26

[46] J. Li, P. Hao, C. Zhang, and M. Dou. Hallucinating faces from thermal infrared images. In *ICIP*, 2008. 26

[47] J. Li, X. Liang, S. Shen, T. Xu, and S. Yan. Scale-aware fast R-CNN for pedestrian detection. *arXiv*, 1510.08160, 2015. 15

[48] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan. Perceptual generative adversarial networks for small object detection. In *CVPR*, 2017. 36

[49] S. Liao, A. K. Jain, and S. Z. Li. A fast and accurate unconstrained face detector. *IEEE transactions on pattern analysis and machine intelligence*, 2016. 26

[50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 13

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *arXiv*, 1512.02325, 2015. 15

[52] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 18, 33, 35, 36

[53] Y. M. Lui, D. Bolme, B. A. Draper, J. R. Beveridge, G. Givens, and P. J. Phillips. A meta-analysis of face recognition covariates. In *IEEE BTAS*, 2009. 4, 5, 6

[54] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016. 16, 18

[55] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv:1606.08921*, 2016. 35, 37

[56] K. Nasrollahi and T. B. Moeslund. Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 2014. 35

[57] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern recognition*, 1993. 35

[58] S. W. Park and M. Savvides. Breaking the limitation of manifold analysis for super-resolution of facial images. In *ICASSP*, 2007. 4

[59] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015. 30

[60] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv:1606.02147*, 2016. 36

[61] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, 2017. 36

[62] X. Peng, J. Hoffman, X. Y. Stella, and K. Saenko. Fine-to-coarse knowledge transfer for low-res image classification. In *ICIP*, 2016. xi, 5, 10

[63] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. *FG*, 2017. 25

[64] S. Razakarivony and F. Jurie. Vehicle detection in aerial imagery. *J. Vis. Comun. Image Represent.*, 2016. 15

[65] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv*, 1506.02640, 2015. 15

[66] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS 2015*, 2015. 13, 14, 15, 17, 21

[67] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 2017. 36, 37

[68] M. S. Ryoo, K. Kim, and H. J. Yang. Extreme low resolution activity recognition with multi-siamese embedding learning. *arXiv:1708.00999*, 2017. 3, 5, 11

[69] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *arXiv:1602.03418*, 2016. 26

[70] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 24, 26

[71] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 24, 27, 31

[72] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen. Classification driven dynamic image enhancement. *arXiv:1710.07558*, 2017. 11

[73] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *arXiv*, 1609.05158, 2016. 16

[74] F. Šroubek and J. Flusser. Resolution enhancement via probabilistic deconvolution of multiple degraded images. *Pattern Recognition Letters*, 2006. 34

[75] J.-C. Su and S. Maji. Adapting models to signal degradation using distillation. 2017. 10

[76] K. Su, Q. Tian, Q. Xue, N. Sebe, and J. Ma. Neighborhood issue in single-frame image super-resolution. In *ICME*, 2005. 4

[77] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014. 26

[78] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 24, 26

[79] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2008. ix, 6, 7

[80] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *ICCV 2011*. 15

[81] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 2004. 26

[82] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *CVPR*, 2016. 35

[83] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang. Studying very low resolution recognition using deep networks. In *CVPR*, 2016. 5, 6, 9, 11

[84] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR 16*. 15

[85] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *CVPR*, 2016. 10, 26, 27, 30

[86] S. Yang, P. Luo, C. C. Loy, and X. Tang. Faceness-net: Face detection through deep facial part responses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 24, 26

[87] G. Ye, M. Pickering, M. Frater, and J. Arnold. A robust approach to super-resolution sprite generation. In *ICIP*, 2005. 34

[88] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 26

[89] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 33, 35

[90] X. Yu and F. Porikli. Ultra-resolving face images by discriminative generative networks. In *European Conference on Computer Vision*, 2016. 26

[91] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014. 26

[92] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 33, 35

[93] Y. Zhou, Y. Qu, Y. Xie, and W. Zhang. Image super-resolution using deep belief networks. In *ICIMCS*, 2014. 35

[94] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV 2014*. 15