

Deep Learning for Computer Vision
Professor Vineeth n Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
Lecture 51
CNNs for Other Image Tasks

In the last lecture of this week, we will very briefly look at a few other applications, such as depth estimation, super resolution and anomaly detection ways in which CNNs can be used for these tasks.

(Refer Slide Time: 00:37)

The slide is titled "Depth Estimation" in red text. On the left side, there are two logos: the NPTEL logo (National Programme on Technology Enhanced Learning) and the IIT Hyderabad logo. In the center, there is a photograph of a white car parked on a city street. A person is standing on the sidewalk to the left of the car. The background shows a modern building with a glass facade. At the bottom left of the slide, there is a small video thumbnail showing the professor, Vineeth N B. At the bottom center, there is a footer with the text "Vineeth N B (IIT-H)" and "57.6 CNNs for Other Image Tasks". At the bottom right, there is a page number "2 / 8".

Taking the application of depth estimation, given a scene such as what you see on this image, there are objects at several depths from the perspective of the camera, you have the road, you have a pillar at 1 corner of the scene, you have the car itself and you also have the person standing on the pavement, each of whom almost act like different layers at different depths from the camera.

(Refer Slide Time: 01:09)

Depth Estimation

(1600, 2400, 3)

Vineeth N B (IIT-H) | 7.6 CNNs for Other Image Tasks | 2 / 8

The question that we would like to ask here is, can we get estimates of depth using just single images? Existing methods for depth estimation, including the human visual system rely on stereo estimates where there are two cameras, but the question we would like to ask now is can you get depth from just a single 2D image, and there have been efforts to do this, we will look at a couple of samples briefly in this lecture.

(Refer Slide Time: 01:41)

Depth Estimation from Single Image¹

Coarse network

Finer network

Input

Refined

¹Eigen et al, Depth map prediction from a single image using a multi-scale deep network, NeurIPS 2014

Vineeth N B (IIT-H) | 7.6 CNNs for Other Image Tasks | 3 / 8

So, one of the earliest efforts of using deep CNNs for depth estimation was in NeurIPS 2014, where given an input a coarse level CNN is used to forward propagate the image and then at the end of the last layer, get a pixel level estimate of the depth for each pixel in the image. This estimate, very similar to the approaches for pose estimation or similar applications that

we have seen, also uses a refinement step to improve the precision of the depth estimate. In this case, how this is done is the same image also with a different filter at a different resolution is passed to a final depth estimation network. And in addition to this network, the coarse estimate of the other network is also passed, and concatenated as an input in an intermediate layer. And together, these two are combined to give a refined depth estimate, which is more accurate.

(Refer Slide Time: 03:00)

Depth Estimation from Single Image: Sample Results

Input Output from coarse network Output from finer network

Credit: Eigen et al, Depth map prediction from a single image using a multi-scale deep network, NeurIPS 2014

Vineeth N B (IIT-J) §7.6 CNNs for Other Image Tasks 4 / 8

Here are examples of this kind of an approach, where given an input, the output of the coarse network looks something like this. And after refinement, it starts becoming more usable.

(Refer Slide Time: 3:13)

Depth Estimation from Single Image: Sample Results

Input Output from coarse network Output from finer network

Credit: Eigen et al, Depth map prediction from a single image using a multi-scale deep network, NeurIPS 2014

Vineeth N B (IIT-J) §7.6 CNNs for Other Image Tasks 4 / 8

Here is another example, given an input image, here is the depth estimate from the coarse network and here is the depth estimate after getting this from the final network. The assumption here is you have a pixel wise estimate of the depth provided to you as ground truth. So you could get a simple L 2 error or a mean square error pixel wise to be able to use as a loss function to train these networks.

(Refer Slide Time: 3:39)

GeoNet²

Consists of rigid structure reconstructor for estimating static scene geometry and non-rigid motion localizer for capturing dynamic objects. Consistency check within any pair of bidirectional flow predictions is adopted for taking care of occlusions and non-Lambertian surfaces

²Yin and Shi, GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose, CVPR 2018

Vineeth N B. (IIT-H) | 7.6 CNNs for Other Image Tasks | 5 / 8

A more contemporary approach for depth estimation, called GeoNet is very popularly used. We will only visit this briefly, since it brings in concepts from vision which we have not covered in detail in this particular course. But this network performs both depth estimation and camera motion estimation and pose estimation in a combined framework, but let us focus on the depth estimation part here.

It assumes that depth can be estimated when you have a sequence of images rather than a single image, which is likely to happen if you are using this for an application say like autonomous driving, you have a camera that is captured as a car drives, and you are going to get a sequence of frames, which can be given as input as a volume, just like how we gave RGB channels as inputs to a CNN so far, this network takes the frames captured over an entire image sequence over time as different channels of an input volume.

Then a depth network estimates depth on each of these channels, and interactions between the depth maps of these temporal frames give an estimate of what is known as rigid flow which is the flow of stationary objects in the scene. And this is then combined with a non-rigid motion localizer which uses optical flow, if you recall, we talked about optical flow in the initial

lectures of this course. So, it uses an optical flow based approach to get a non-rigid motion localizer, this is for dynamic content in the scene. And finally, combining the rigid flow and the non-rigid motion localizer, the final flow or the depth is predicted for each of these for this entire scene. And at the end, a consistency check is done to check for whether the flow predictions take care of occlusions and even non Lambertian surfaces. For more details, you can see this paper. This was published in CVPR 2018.

(Refer Slide Time: 06:02)

Super-resolution: Do you see a difference?

NPTEL

Vineeth N B (IIT-JH) [7.6 CNNs for Other Image Tasks] 6 / 8

Another task, which is growingly popular with deep neural networks is the task of super resolution. Can you see a difference in these two images? There is no semantic difference but there is a perceptual difference in terms of the resolution of the right image over the left image. And this is used extensively to improve the resolution of content.

(Refer Slide Time: 06:31)

Super-resolution

Credit: LG

Vineeth N B (IIT-H) 7.6 CNNs for Other Image Tasks 6 / 8

Today, televisions have already started enhancing the resolution of the image presented on the screen using such approaches. Here is an example of an LG screen, which in its settings allows you to choose an AI enhanced resolution up-scaling option, which in fact does super resolution to get the final image on the screen, so it is ultra HD, where you use CNN based approaches to super resolve images and get a better resolution on the screen. So how do you do super resolution, there are multiple ways in which this can be done.

(Refer Slide Time: 7:12)

Super-resolution using CNNs

Low-resolution image (input) $f_1 \times f_1$ n_1 feature maps of low-resolution image $f_2 \times f_2$ Non-linear mapping n_2 feature maps of high-resolution image $f_3 \times f_3$ High-resolution image (output)

Patch extraction and representation Reconstruction

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3$$

Credit: Dong, Chao, et al, Image Super-resolution using Deep Convolutional Networks, IEEE Trans on PAMI, 2015

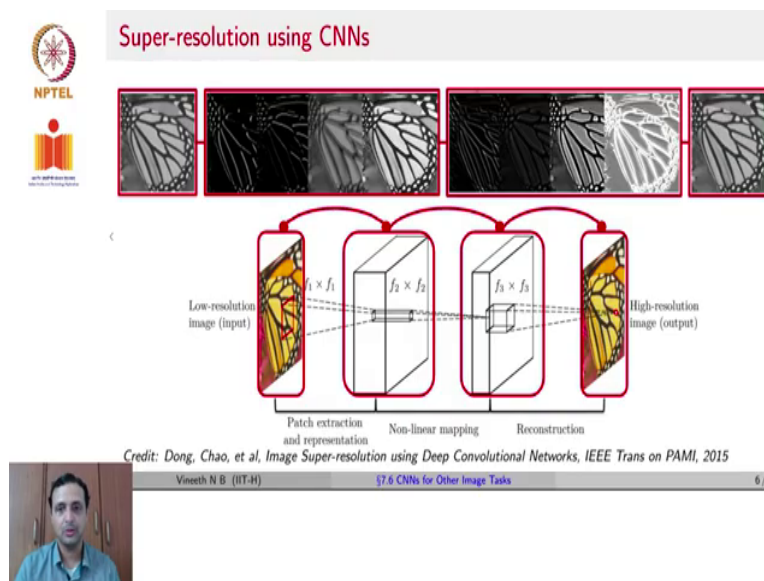
Vineeth N B (IIT-H) 7.6 CNNs for Other Image Tasks 6 / 8

We will talk about a CNN based approach now. Later in this course, we will also talk about generative based approaches, or what are known as Gann based approaches to super resolve image content. Let us see how this is done using CNNs. You have a seemingly low resolution

image that is provided as input to a set of convolutional layers, which use the idea that a patch around a pixel can be used to improve the resolution of the image around that pixel in the output image. Let us see this in more detail. You have a low resolution image input, which is passed on to get n_1 feature maps of the low resolution image. So you have $F_1(Y) = \max(0, W_1 * Y + B_1)$ where y we assume is the input.

And using that $F_1(Y)$, you are going to forward propagate that to n_2 feature maps of the next layer, which gives you $F_2(Y)$ which is a relu over a linear layer of the $F_1(Y)$ feature maps. And finally, these $F_2(Y)$'s are used over a linear layer over a set of weights to get the final reconstruction of the high resolution image. Assuming that you have data of high resolution versions of low resolution images, you can use a pixel wise loss to train this entire CNN. How does this work?

(Refer Slide Time: 8:45)

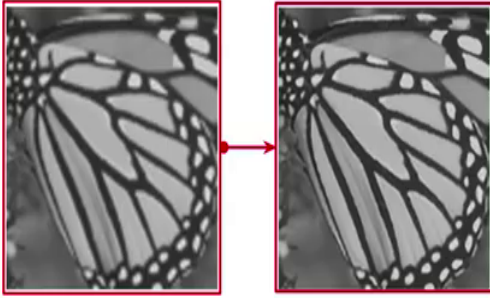


Given this input image of the butterfly wing, the feature maps of the first convolutional layer here obtain feature maps such as these, some of them recognize edges, slightly varying textures, or so on and so forth. These feature maps are passed on to the next convolutional layer, whose feature maps may look something like this. And finally, all of this information together, is put together to get the final output.

(Refer Slide Time: 9:20)

NPTEL

Super-resolution using CNNs




Credit: Dong, Chao, et al, Image Super-resolution using Deep Convolutional Networks, IEEE Trans on PAMI, 2015

Vineeth N B. (IIT-J)

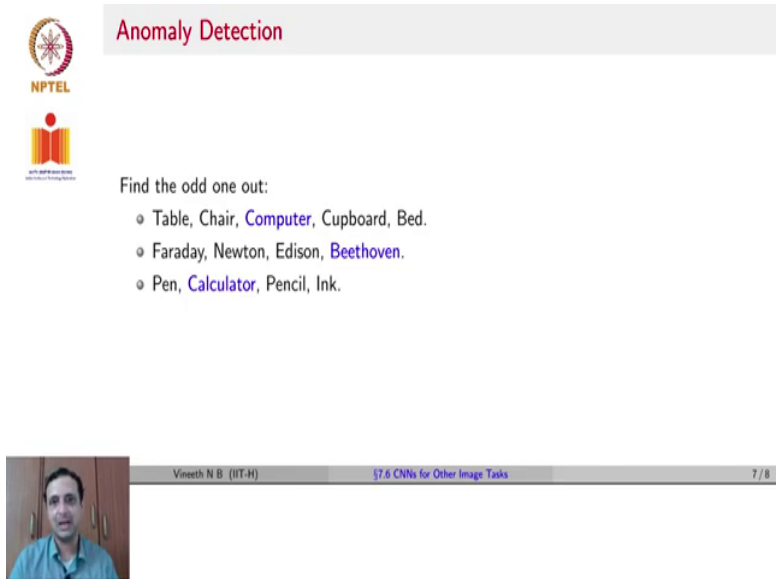
57.6 CNNs for Other Image Tasks

6 / 8



So you can see that given this input to the super resolution CNN, you get an output, where the resolution and the contrast looks far better from a perceptual point of view.

(Refer Slide Time: 9:35)



Anomaly Detection

Find the odd one out:

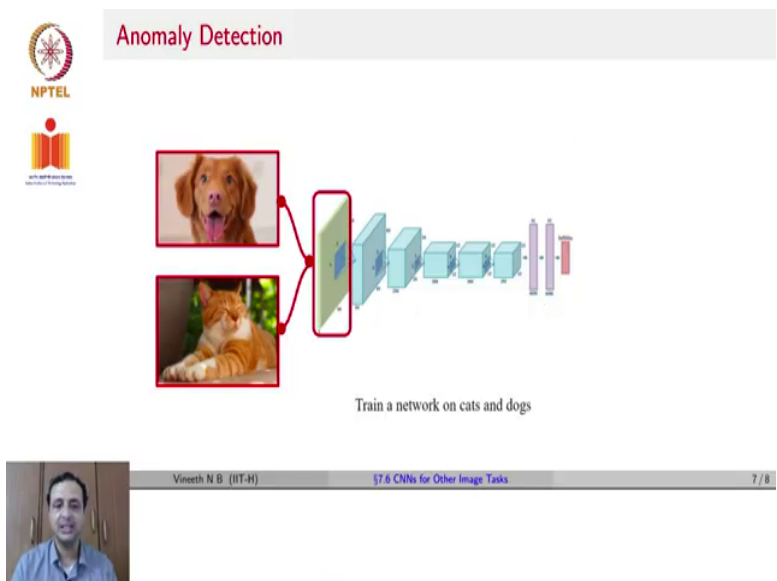
- Table, Chair, **Computer**, Cupboard, Bed.
- Faraday, Newton, Edison, **Beethoven**.
- Pen, **Calculator**, Pencil, Ink.

Vineeth N B (IIT-J) 57.6 CNNs for Other Image Tasks 7/8

A third task that we will briefly describe to close the lectures in this week is the task of anomaly detection. It is a need in many applications today, where given a set of inputs or set of data points, one may need to find out which of these is an outlier, or anomalous with respect to the distribution that we are handling.

Generally speaking, given, say keywords such as table chair, computer cupboard, bed, or Faraday, Newton, Edison, Beethoven, or pen, calculator, pencil, ink, you are looking to find the odd one out. Computer in the first case, Beethoven in the second case, and calculator in the third case. But how do we do this with images using a CNN is what we talk about now.

(Refer Slide Time: 10:30)



Anomaly Detection

Train a network on cats and dogs

Vineeth N B (IIT-J) 57.6 CNNs for Other Image Tasks 7/8

So our goal now is to find out an out of distribution image from a given training data set. So an image comes, which is not in the training data set. How do you now distinguish this from what you have in the dataset? Let us consider a known CNN architecture. Let us assume this neural network is trained on, say cats and dogs.

(Refer Slide Time: 10:53)

Anomaly Detection

Flawed by design

During inference, provide the network with an out of distribution image

Vineeth N B. (IIT-H) 57.6 CNNs for Other Image Tasks 7/8

Now, at test time, you now get an image that does not belong to the classes that the network was trained on. We call such an image as an out of distribution image. If you simply try to classify this into a cat, or a dog, this is not really going to give you a useful answer, because that is the label space that you have in the neural network. But that is not going to give you a useful answer. And trying to give probabilities for cat and dog is flawed by design. So what do we do?

(Refer Slide Time: 11:28)

The slide is titled "Anomaly Detection" and features the NPTEL logo. It displays a diagram of a neural network with several layers of nodes. A red box highlights the output layer, which is connected to a mathematical equation for the softmax function with temperature:
$$S_i(\mathbf{x}; T) = \frac{\exp(f_i(\mathbf{x}))}{\sum_{j=1}^N \exp(f_j(\mathbf{x}))}$$
. Below the diagram, there is a small video feed of the speaker, the name "Vineeth N B. (IIT-H)", the course title "57.6 CNNs for Other Image Tasks", and the slide number "7/8".

We are going to look at the softmax activation function that you use in the last layer. And a recent work, which tried to propose an interesting approach for detecting such anomalies or out of distribution images proposed that you could use the notion of what is known as temperature, which is this new added component in the softmax activation function.

What does temperature do? T is a constant that you provide as input to the softmax activation function. As you can see here, T scales down the inputs that are provided to the exponential function here. How does this help? By scaling things down, the exponential function will now be able to separate these values that you get for different classes better. This idea of using temperature in softmax, is used in many other kinds of applications and we will see some of them a little later in this course.

(Refer Slide Time: 12:39)

Anomaly Detection³

$$S_i(\mathbf{x}; T) = \frac{\exp(f_i(\mathbf{x})/T)}{\sum_{j=1}^N \exp(f_j(\mathbf{x})/T)}$$

$$\hat{\mathbf{x}} = \mathbf{x} - \varepsilon \text{sign}(-\nabla_{\mathbf{x}} \log S_y(\mathbf{x}; T))$$

$$g(\mathbf{x}; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if } \max_i p(\hat{\mathbf{x}}; T) \leq \delta, \\ 0 & \text{if } \max_i p(\hat{\mathbf{x}}; T) > \delta. \end{cases}$$
 in-distribution
out-of-distribution

³Liang et al, Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks, ICLR 2018
Vineeth N B (IIT-H) 17.6 CNNs for Other Image Tasks 7/8



In addition to temperature, there is another contribution this approach makes, this work was published in 2018. Assuming that the model predicted a dog as the output for a given image, you do not know if the image was really of a dog or an out of distribution image. You now consider the gradient of the loss function with respect to this input.

Consider the sine of the negative gradient, and now add an epsilon perturbation, add or subtract based on the sign and epsilon perturbation to the input. Why do we do this? We expect that if this was really a dog image, adding this perturbation would help us recognize better that it is a dog, or the softmax outputs would make it even clearer that it is a dog by taking the log softmax the gradient of the log softmax and its sine in this perturbation calculation.

On the other hand, if it was an out of distribution image, and you added this perturbation, now the softmax outputs would get a bit more confused. This simple intuition is now used to reconstruct an image based on this perturbation, a small perturbation, you provide that as input to the neural network again, and now you look at its output and see if the output was greater than a threshold.

You say that it is not an anomaly, it is perhaps an image of a dog itself. But if the output now is less than a threshold, because that is when the confusion among the softmax would increase it is an anomaly. A simple approach without changing much in the entire pipeline of the neural network or the architecture of the training, but it actually works well to detect an out of distribution image or an anomaly in practice.


(Refer Slide Time: 14:37)



Homework

Readings

- Depth Estimation : <https://paperswithcode.com/task/depth-estimation>
- Super Resolution : <https://paperswithcode.com/task/super-resolution>
- Anomaly Detection : <https://paperswithcode.com/task/anomaly-detection>



Vineeth N B (IIT-H) 57.6 CNNs for Other Image Tasks 8 / 8

For more details, please read these links for depth estimation, super resolution or anomaly detection.