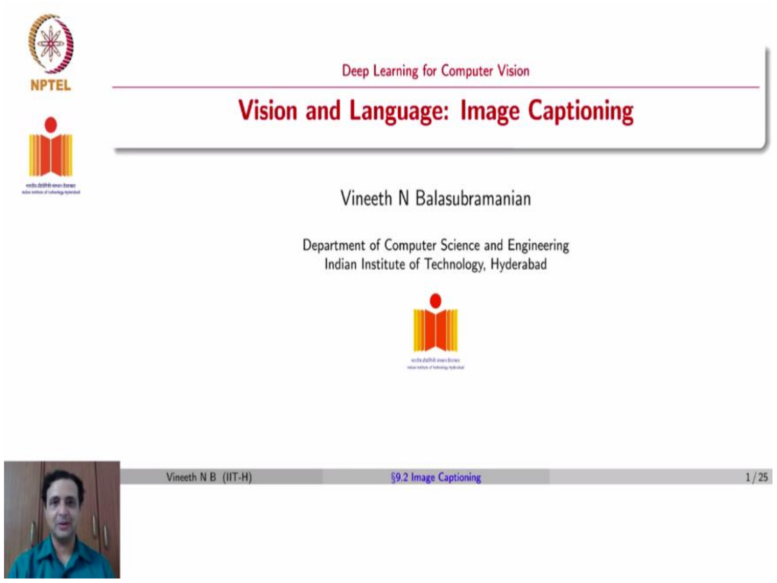
Deep Learning for Computer Vision

Professor Vineeth N Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Lecture - 57
Vision and Language: Image Captioning

(Refer Slide Time: 00:13)



Deep Learning for Computer Vision

Vision and Language: Image Captioning

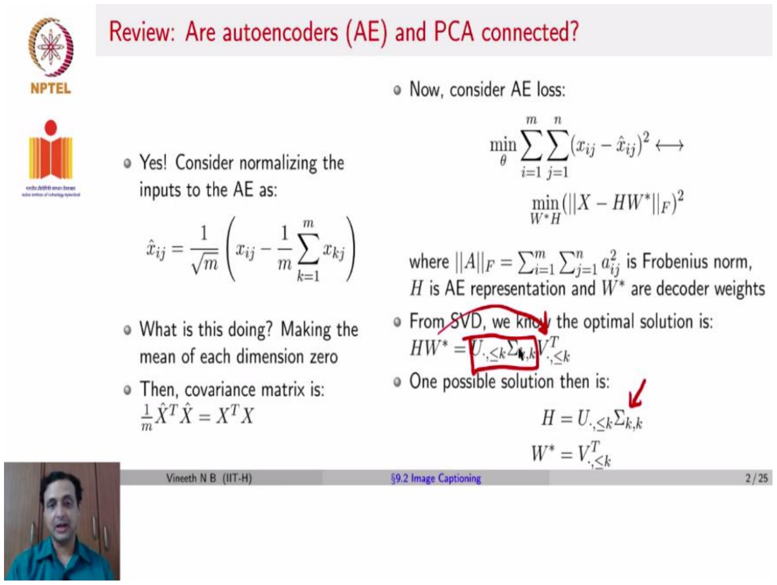
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Vineeth N B (IIT-H) §9.2 Image Captioning 1 / 25

We will now focus on one of the tasks that we discussed in the previous lecture, which is Image Captioning.

(Refer Slide Time: 00:25)



Review: Are autoencoders (AE) and PCA connected?

- Yes! Consider normalizing the inputs to the AE as:

$$\hat{x}_{ij} = \frac{1}{\sqrt{m}}\left(x_{ij} - \frac{1}{m}\sum_{k=1}^{m} x_{kj}\right)$$

- What is this doing? Making the mean of each dimension zero
- Then, covariance matrix is: $\frac{1}{m}\hat{X}^T\hat{X} = X^TX$
- Now, consider AE loss:

$$\min_{\theta}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - \hat{x}_{ij})^2 \longleftrightarrow$$

$$\min_{W^*H}(||X - HW^*||_F)^2$$

where $||A||_F = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2$ is Frobenius norm, H is AE representation and W^* are decoder weights

- From SVD, we know the optimal solution is: $HW^* = \boxed{U_{\cdot,\le k}\Sigma_{k,k}}V_{\cdot,\le k}^T$
- One possible solution then is:

$$H = U_{\cdot,\le k}\Sigma_{k,k}$$

$$W^* = V_{\cdot,\le k}^T$$

Vineeth N B (IIT-H) §9.2 Image Captioning 2 / 25

We left behind one question from the previous lecture, which was our Autoencoders, which we understood as a dimensionality reduction technique. Similar to PCA, Principal Component Analysis in some manner. The answer is yes, let us try to understand the connection. If you normalize your data to the autoencoder in a particular way, where x hat, which is then normalized data is given by 1 by root m into x minus mean of that dimension under consideration.

So, what is this Normalization doing? It is making the mean of each dimension to be 0. It is a standard normalization step. And when you do this kind of normalization, the covariance matrix is given by $X^T X$, which is the covariance is given by $\frac{1}{m}\hat{X}^T\hat{X}$. This is because of the $\frac{1}{\sqrt{m}}$ that you have here when you compute your $\hat{x}$. Given this pre-processing, let us try to look at the Autoencoder loss.

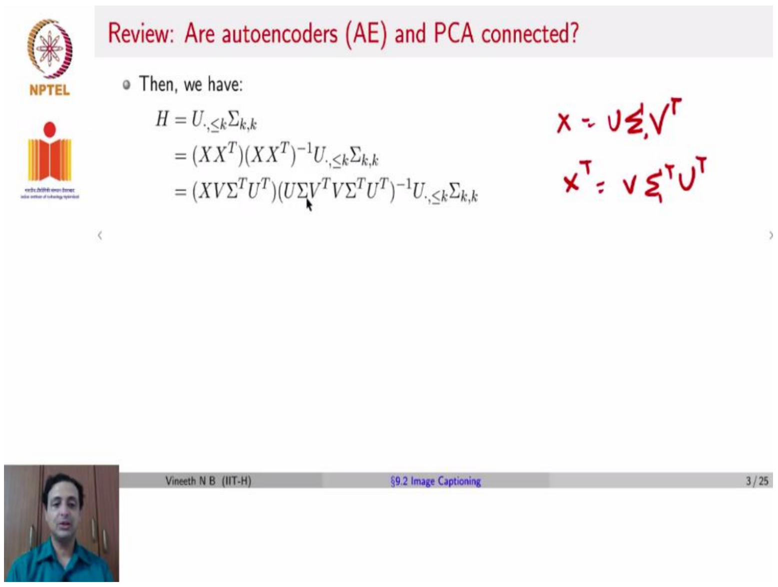
The autoencoder loss is let us say the parameters of your autoencoder are summarized as theta is the mean squared error between your reconstruction and the original input. One can also write this loss in terms of x and can write it as $(\|X - HW^*\|_F)^2$. In this case, the Frobenius norm is given by summation along each of the dimensions, aij whole square is the squared sum of all elements of a matrix.

H here is the representation of the autoencoder in the middle layer, the bottleneck layer or the context vector, and W^* is the weights of your decoder. So, W star into H is the reconstructed output from that intermediate representation. X is the original input. So both of these are equivalent representations for the loss function. Why did we write it this way? Let us see a bit more.

We know that for a loss function written this way, HW^* using SVD can be written as $U\Sigma V^T$, where you consider only the k columns of U and V^T where you have a non 0 rank a non 0 singular value. So, that is what we are writing this HW^* as this comes from the SVD decomposition of x.

So if this was indeed the solution, then one factoring could be H could be the first two terms here. So that would be U, Σ. And W could be V here. This is one possible factoring and lets us consider this to be one of the outcomes of the neural network and move forward with this outcome. So if you considered H to be U, Σ, or considering only the k columns.

(Refer Slide Time: 04:03)



Review: Are autoencoders (AE) and PCA connected?

- Then, we have:

$$H = U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XX^T)(XX^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XV\Sigma^T U^T)(U\Sigma V^T V\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$

$X = U\Sigma V^T$

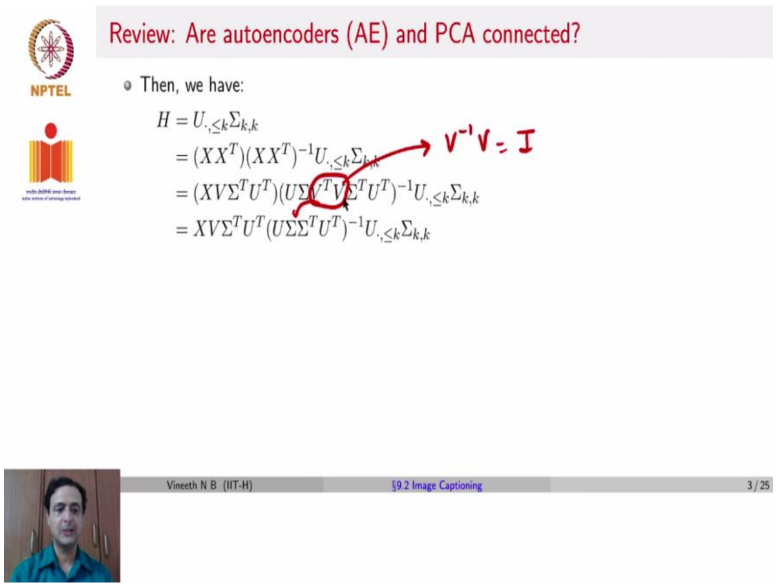
$X^T = V\Sigma^T U^T$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

Then you have H which can be written now as let us multiply by $(XX^T)(XX^T)^{-1}$, which we know is identity. We are just taking the expansion of H and multiplying by identity which is given by $(XX^T)(XX^T)^{-1}$. Now we know from SVD of Z, we can write X^T as V. So remember, if we have $X = U\Sigma V^T$, then $X^T = V\Sigma U^T$.

That is what we are writing here. And $X = U\Sigma V^T$ and $X^T = (V\Sigma U^T)^{-1}$.

(Refer Slide Time: 05:02)



Review: Are autoencoders (AE) and PCA connected?

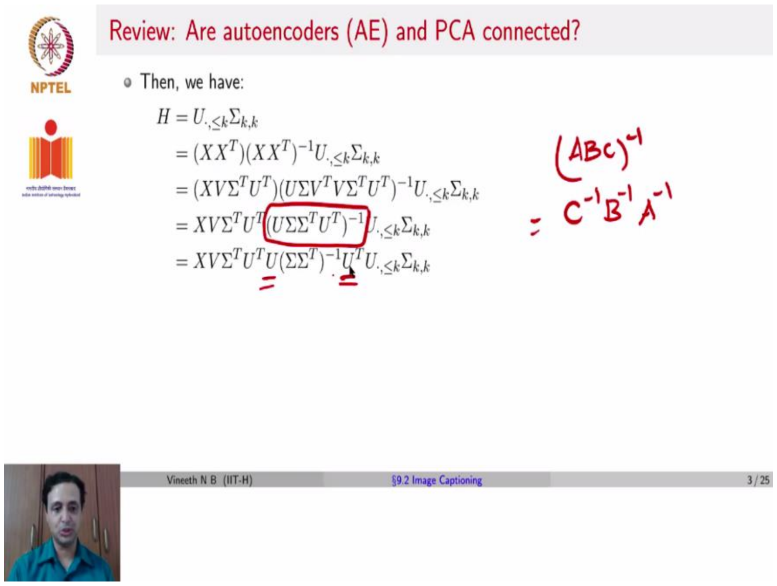
- Then, we have:

$$H = U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XX^T)(XX^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XV\Sigma^T U^T)(U\Sigma \boxed{V^T V}\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \rightarrow V^{-1}V = I$$
$$= XV\Sigma^T U^T(U\Sigma\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

Now, by looking at this, you can make out that again from the definition of SVD V^TV is would be identity because in SVD V is an orthogonal, orthonormal matrix or orthogonal matrix, where $V^T = V^{-1}$. Hence, $V^TV = V^{-1}V$, which means it should be I. So, when this term here is $V^{-1}V$, which is I, because V is an orthogonal matrix. So, that collapses and you are left with $U\Sigma\Sigma^TU^T$.

(Refer Slide Time: 05:43)



Review: Are autoencoders (AE) and PCA connected?

- Then, we have:

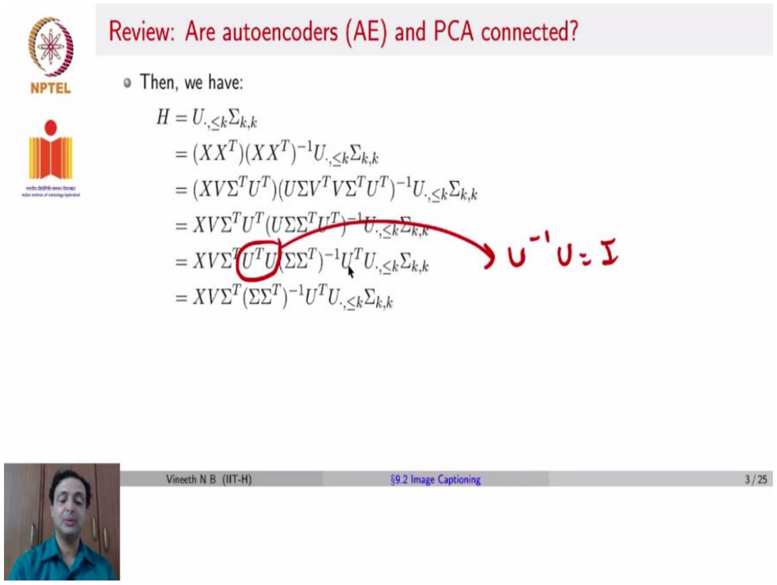
$$H = U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XX^T)(XX^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= (XV\Sigma^T U^T)(U\Sigma V^T V\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= XV\Sigma^T U^T\boxed{(U\Sigma\Sigma^T U^T)^{-1}}U_{\cdot,\leq k}\Sigma_{k,k}$$
$$= XV\Sigma^T \underline{U^T U}(\Sigma\Sigma^T)^{-1}\underline{U^T}U_{\cdot,\leq k}\Sigma_{k,k}$$

$$(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

Now, you can also write this as the first U comes out here and you have, $(\Sigma\Sigma^T)^{-1}U^T$. How did this happen? This comes from the inverse of this matrix here. Remember, if you have ABC three matrices whole inverse, that is given by $C^{-1}B^{-1}A^{-1}$. Now, $(U^T)^{-1}$ will be U itself, $(\Sigma\Sigma^T)^{-1}$ stays that way. And $U^{-1} = U^T$. Once again, you can interchange inverse and transpose because U is an orthogonal matrix when you do SVD decomposition.

(Refer Slide Time: 06:35)



Review: Are autoencoders (AE) and PCA connected?

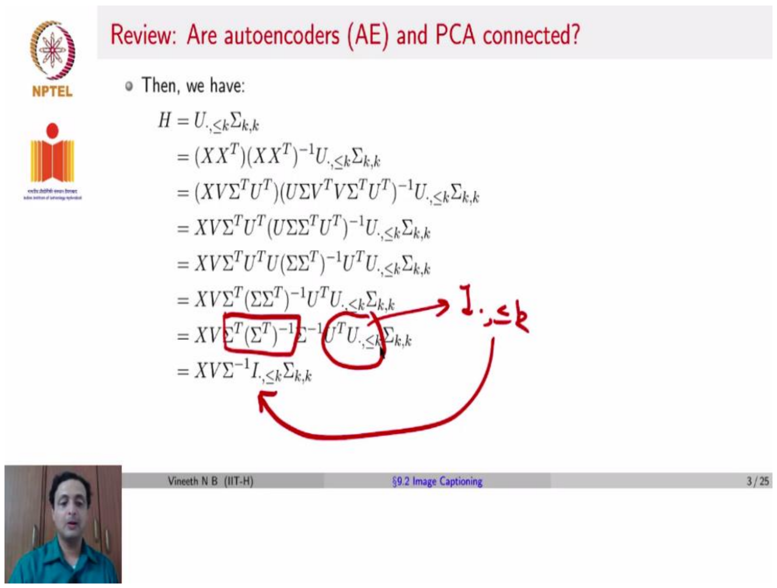
- Then, we have:

$$
\begin{aligned}
H &= U_{\cdot,\leq k}\Sigma_{k,k} \\
&= (XX^T)(XX^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\
&= (XV\Sigma^T U^T)(U\Sigma V^T V\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\
&= XV\Sigma^T U^T(U\Sigma\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\
&= XV\Sigma^T \boxed{U^T U}(\Sigma\Sigma^T)^{-1}U^T U_{\cdot,\leq k}\Sigma_{k,k} \xrightarrow{\quad} U^{-1}U = I \\
&= XV\Sigma^T(\Sigma\Sigma^T)^{-1}U^T U_{\cdot,\leq k}\Sigma_{k,k}
\end{aligned}
$$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

Now, this can be collapsed as $U^T U$ because it is orthogonal would be the same as $U^{-1}U$, which is the identity matrix. So, that collapses, and you are left with the rest of the terms.

(Refer Slide Time: 06:48)



Review: Are autoencoders (AE) and PCA connected?

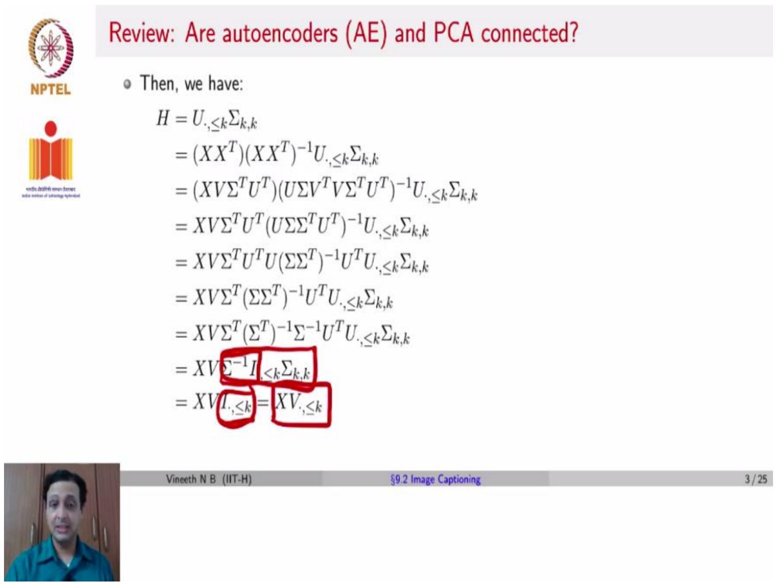
- Then, we have:

$$
\begin{aligned}
H &= U_{.,\leq k}\Sigma_{k,k}\\
&= (XX^T)(XX^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= (XV\Sigma^TU^T)(U\Sigma V^TV\Sigma^TU^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^TU^T(U\Sigma\Sigma^TU^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^TU^TU(\Sigma\Sigma^T)^{-1}U^TU_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^T(\Sigma\Sigma^T)^{-1}U^TU_{.,\leq k}\Sigma_{k,k} \rightarrow I_{.,\leq k}\\
&= XV\boxed{\Sigma^T(\Sigma^T)^{-1}}\Sigma^{-1}\boxed{U^TU_{.,\leq k}}\Sigma_{k,k}\\
&= XV\Sigma^{-1}I_{.,\leq k}\Sigma_{k,k}
\end{aligned}
$$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

And once again here,$(\Sigma\Sigma^T)^{-1} = (\Sigma^T)^{-1}\Sigma^{-1}$. And that is what we are writing here. Now, you have a $\Sigma\Sigma^T(\Sigma\Sigma^T)^{-1}$, which would be identity and collapse. So, you will be left with $XV\Sigma^{-1}U^TU$, once again U^TU, where U is constrained by the first k columns would be given by I constrained to the first k columns which are what we write here.

(Refer Slide Time: 07:35)



Review: Are autoencoders (AE) and PCA connected?

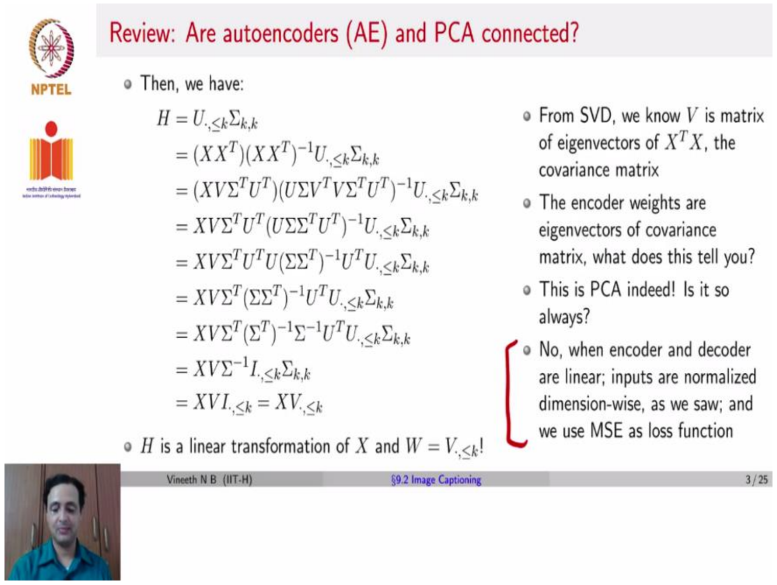
- Then, we have:

$$
\begin{aligned}
H &= U_{.,\leq k}\Sigma_{k,k}\\
&= (XX^T)(XX^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= (XV\Sigma^TU^T)(U\Sigma V^TV\Sigma^TU^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^TU^T(U\Sigma\Sigma^TU^T)^{-1}U_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^TU^TU(\Sigma\Sigma^T)^{-1}U^TU_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^T(\Sigma\Sigma^T)^{-1}U^TU_{.,\leq k}\Sigma_{k,k}\\
&= XV\Sigma^T(\Sigma^T)^{-1}\Sigma^{-1}U^TU_{.,\leq k}\Sigma_{k,k}\\
&= XV\boxed{\Sigma^{-1}I}\boxed{_{.,\leq k}\Sigma_{k,k}}\\
&= XV\boxed{I_{.,\leq k}} = \boxed{XV_{.,\leq k}}
\end{aligned}
$$

Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

So, with that, we go on to the last step, where $\Sigma^{-1}I$ constrained to the first k columns would be sigma inverse constrained to the first k columns and $\Sigma^{-1}\Sigma_k^{-1}$,k would be I constrained to the first k columns, this can be written as X into V constrained to the first k columns. Why is this derivation important?

(Refer Slide Time: 08:09)



Review: Are autoencoders (AE) and PCA connected?

- Then, we have:

$$\begin{aligned} H &= U_{\cdot,\leq k}\Sigma_{k,k} \\ &= (XX^T)(XX^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\ &= (XV\Sigma^T U^T)(U\Sigma V^T V\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\ &= XV\Sigma^T U^T(U\Sigma\Sigma^T U^T)^{-1}U_{\cdot,\leq k}\Sigma_{k,k} \\ &= XV\Sigma^T U^T U(\Sigma\Sigma^T)^{-1}U^T U_{\cdot,\leq k}\Sigma_{k,k} \\ &= XV\Sigma^T(\Sigma\Sigma^T)^{-1}U^T U_{\cdot,\leq k}\Sigma_{k,k} \\ &= XV\Sigma^T(\Sigma^T)^{-1}\Sigma^{-1}U^T U_{\cdot,\leq k}\Sigma_{k,k} \\ &= XV\Sigma^{-1}I_{\cdot,\leq k}\Sigma_{k,k} \\ &= XVI_{\cdot,\leq k} = XV_{\cdot,\leq k} \end{aligned}$$

- H is a linear transformation of X and $W = V_{\cdot,\leq k}$!

- From SVD, we know V is matrix of eigenvectors of X^TX, the covariance matrix
- The encoder weights are eigenvectors of covariance matrix, what does this tell you?
- This is PCA indeed! Is it so always?
- No, when encoder and decoder are linear; inputs are normalized dimension-wise, as we saw; and we use MSE as loss function

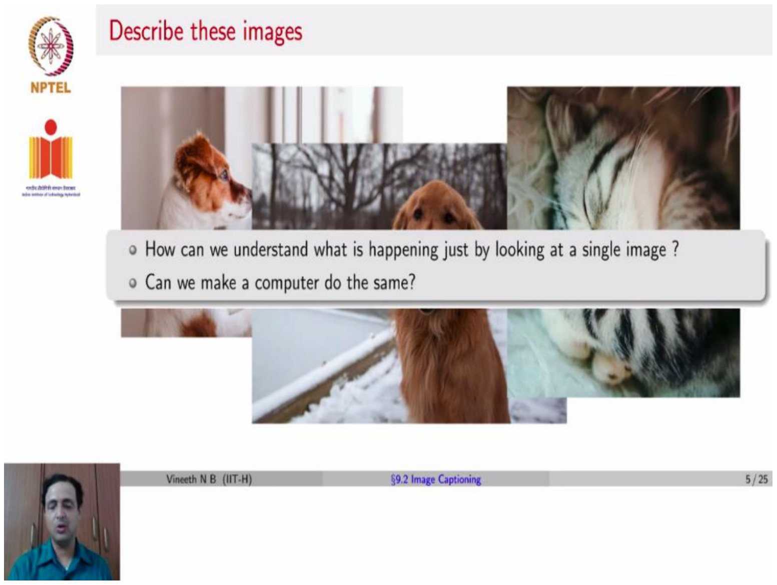
Vineeth N B (IIT-H) §9.2 Image Captioning 3 / 25

This now tells us that H is a linear transformation of X and W, which is in this case, V constrained to the first k columns. What does this mean? From SVD? We know that given a matrix X, and it is SVD - $U\Sigma V^T$ we know that V is a matrix of Eigenvectors of X^TX. Go back and see assignment 0 in the course and refresh the basics if you are not aware of this.

So, which means our encoder weights which are given by V are the Eigenvectors of the covariance matrix. What does this tell you? It should remind you of PCA where the projection matrix is the Eigenvectors of the covariance matrix. So, are our auto encoders always a variant of PCA, or are they do they become PCA? Once you learn the model, not really, this happens when your encoder and decoder are linear.

Remember, in this entire example, we did not talk about having any activation functions in your encoder or decoder. And you have to normalize it, the inputs the way we saw, and you have to use the mean squared error loss function. Under these conditions, the mapping learns by the encoder of autoencoders is indeed the projection matrix of principal component analysis.

(Refer Slide Time: 10:01)



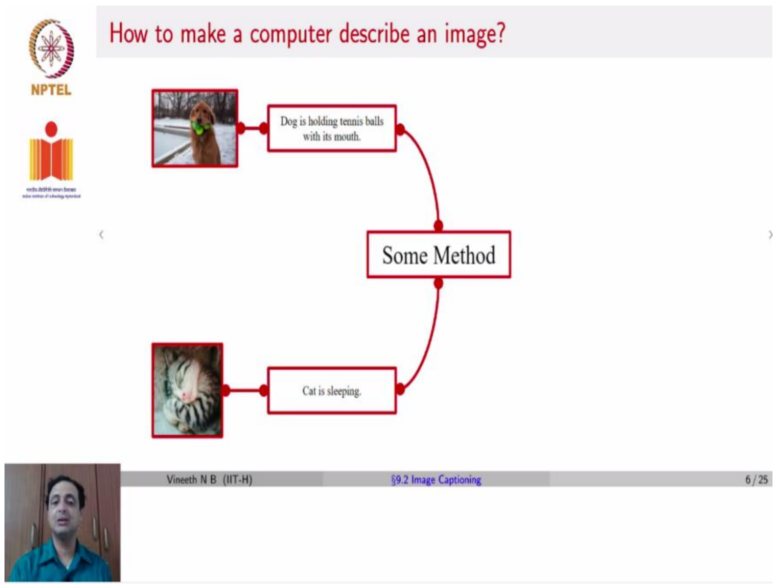
Describe these images

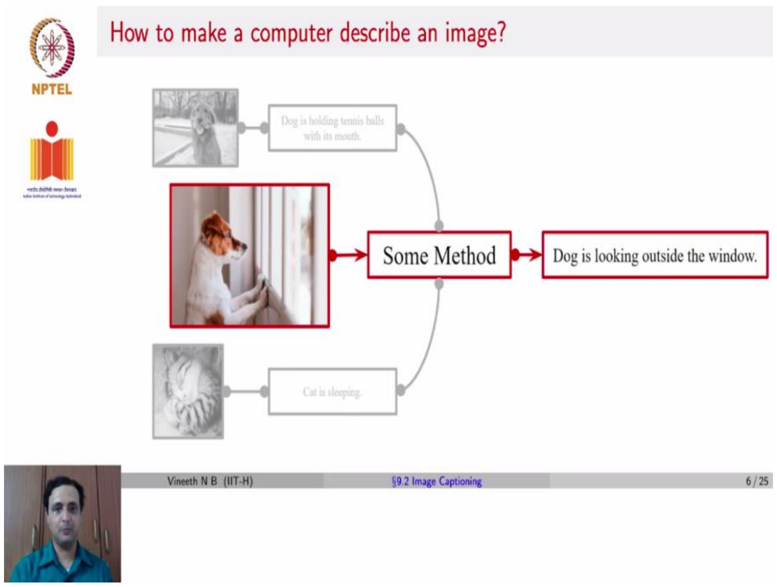
- How can we understand what is happening just by looking at a single image ?
- Can we make a computer do the same?

Vineeth N B (IIT-H) §9.2 Image Captioning 5 / 25

Having answered this question, let us come back to the topic of this lecture, which is Image Captioning. Let us say we wanted to describe an image such as what you see here, or what you see here, or what you see here, or what you see here. As humans, we can describe these images in words. How can we ensure a computer or a Machine Learning model can do the same? How do we go about this?

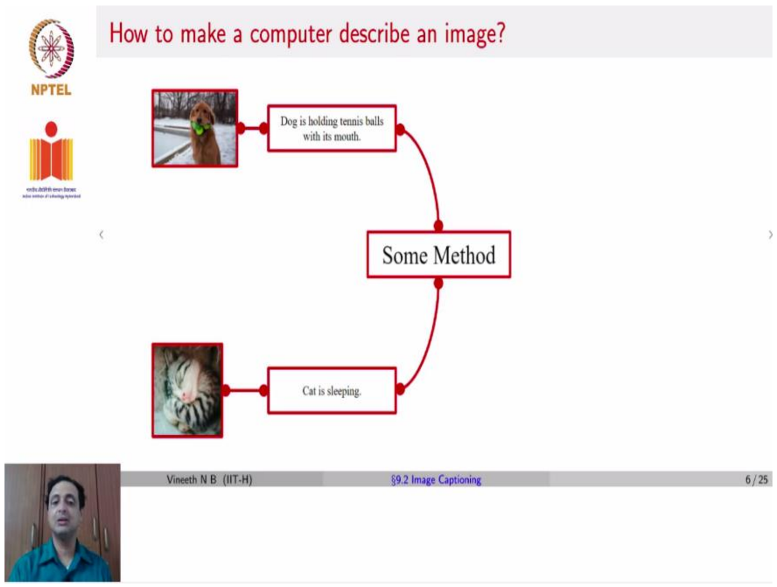
(Refer Slide Time: 10:34)


How to make a computer describe an image?
Dog is holding tennis balls with its mouth.
Some Method
Cat is sleeping.
Vineeth N B (IIT-H)
§9.2 Image Captioning
6 / 25


How to make a computer describe an image?
Dog is holding tennis balls with its mouth.
Some Method
Dog is looking outside the window.
Cat is sleeping.
Vineeth N B (IIT-H)
§9.2 Image Captioning
6 / 25

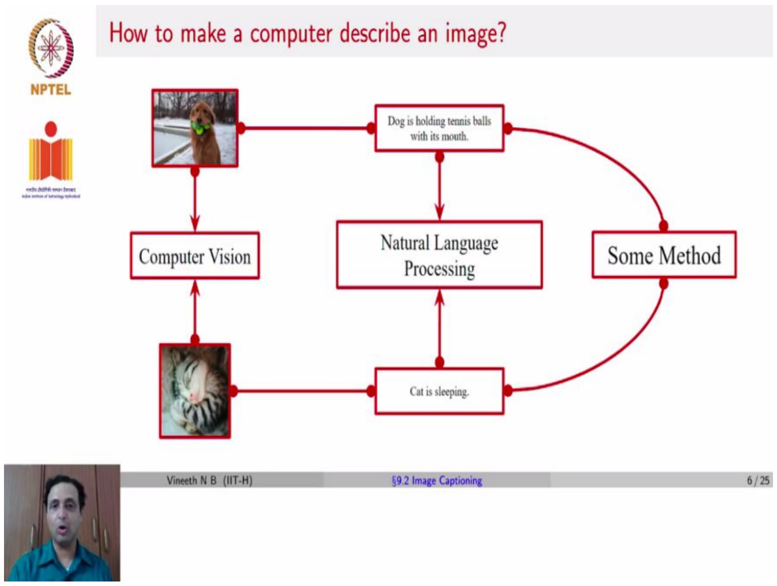
Ideally, we need some method, which can work on the images and give us this output. So given some images, and given some captions, which describe those images, we give them as input to some method, which now takes a new image and gives the output dog is looking outside the window.

(Refer Slide Time: 11:02)



How to make a computer describe an image?

Vineeth N B (IIT-H) §9.2 Image Captioning 6 / 25



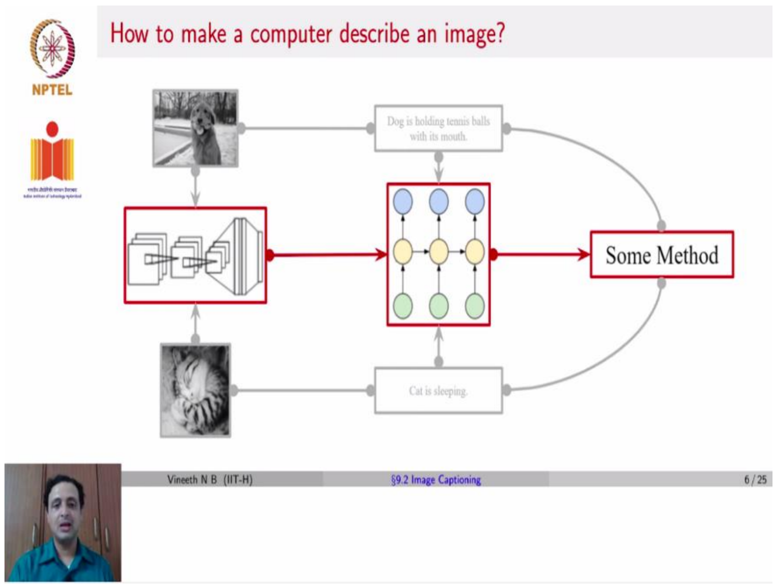
How to make a computer describe an image?

Dog is holding tennis balls with its mouth.
Computer Vision
Natural Language Processing
Some Method
Cat is sleeping.

Vineeth N B (IIT-H) §9.2 Image Captioning 6 / 25

Now, how do we go about coming up with this method, we look at the images and use Vision-based Deep Learning models. We look at the text and use NLP-based deep learning models, which would be RNNs in this case. So in this case, the computer vision part would be replaced by CNNs.

(Refer Slide Time: 11:22)



How to make a computer describe an image?

The NLP part would be replaced by RNNs and you combine them to get your final output as the caption.

(Refer Slide Time: 11:38)

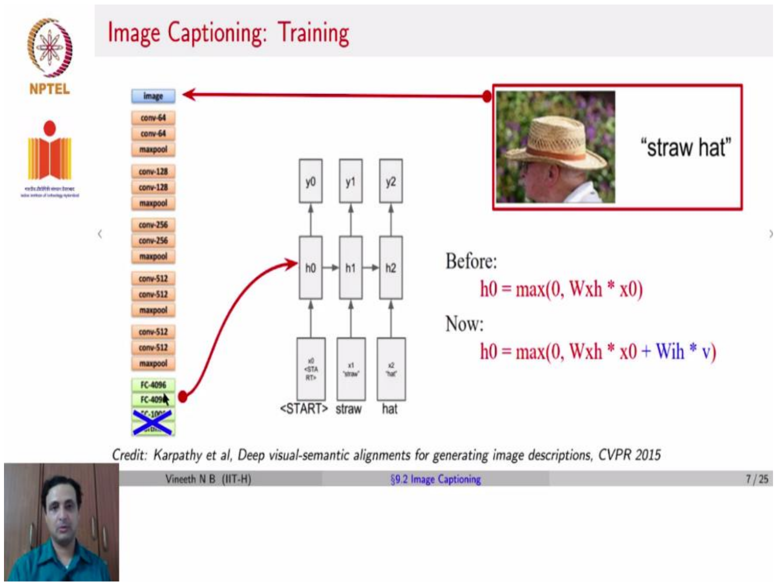


Image Captioning: Training

Before:

h0 = max(0, Wxh * x0)

Now:

h0 = max(0, Wxh * x0 + Wih * v)

Credit: Karpathy et al, Deep visual-semantic alignments for generating image descriptions, CVPR 2015

Let us see each of these components in detail. Let us say we have the training face of such a model. So you have an image. And you know, the corresponding caption is a straw hat. How do we train a relevant model? You take the image and give this image as input to a CNN and you take the text and you use that to train an RNN. However, it is important here to be able to learn the interactions between your CNN and RNN between the image and the text.

How do we do that? We briefly mentioned this in the previous lecture, you discard the last classification layers of your CNN, you take a representation that you have as the output of the CNN and you give that also as input to the RNN. So what, what happens to the RNN now, earlier, your RNN was something like this, where the hidden state H naught was given by say if you use the relu.

Then it would have a max of 0 commas some weight matrix W xh into your input, say x naught. But now, you are going to have H naught to be max of 0 that comes from a relu activation function. A weight matrix W xh into x naught plus a new weight matrix W ih into V where this V vector comes from the CNNs output.

(Refer Slide Time: 13:24)

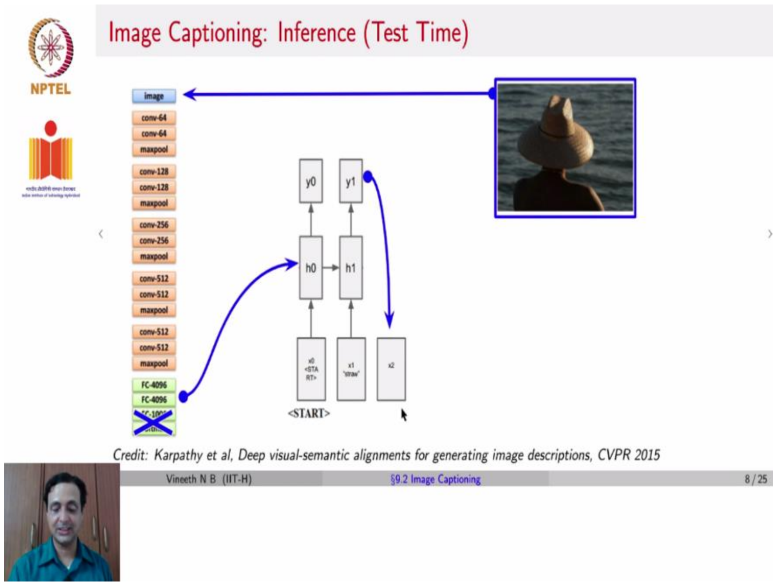


Image Captioning: Inference (Test Time)

Credit: Karpathy et al, Deep visual-semantic alignments for generating image descriptions, CVPR 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 8 / 25

What happens at test time, given a different image, this image as is given as input to the CNN and the CNNs FC seven representation is given as input to an RNN along with a start token, remember, for an RNN to start the output process, you have to give some starting token, there is no word there, you would just consider in your vocabulary, that in addition to all of the words in your vocabulary, you also have two words known as start token and end token.

The start token is the input that you would give to the first-time step of the RNN. So the RNN receives the start token as well as the FC seven representation of the input image. And together these result in a certain output, y naught. And then y naught is then used as x1, to, to generate a new word y1, and y1 is given as input x2

(Refer Slide Time: 14:42)

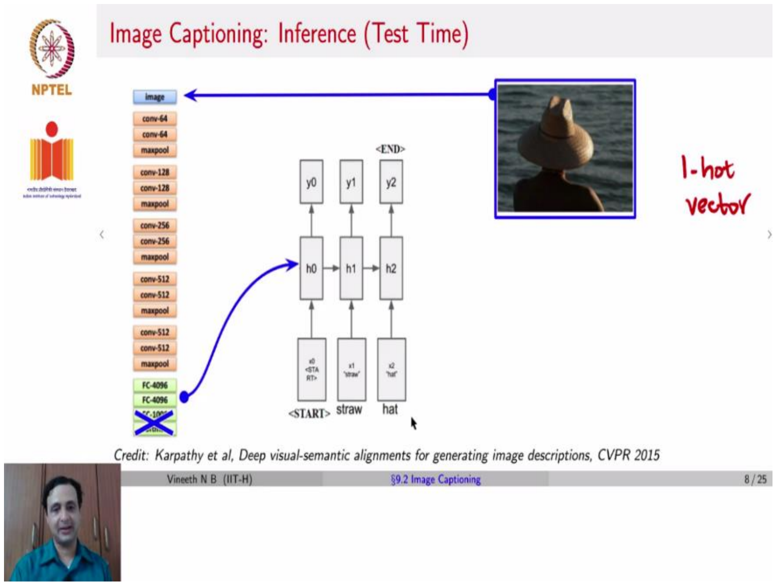


Image Captioning: Inference (Test Time)

Credit: Karpathy et al, Deep visual-semantic alignments for generating image descriptions, CVPR 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 8 / 25

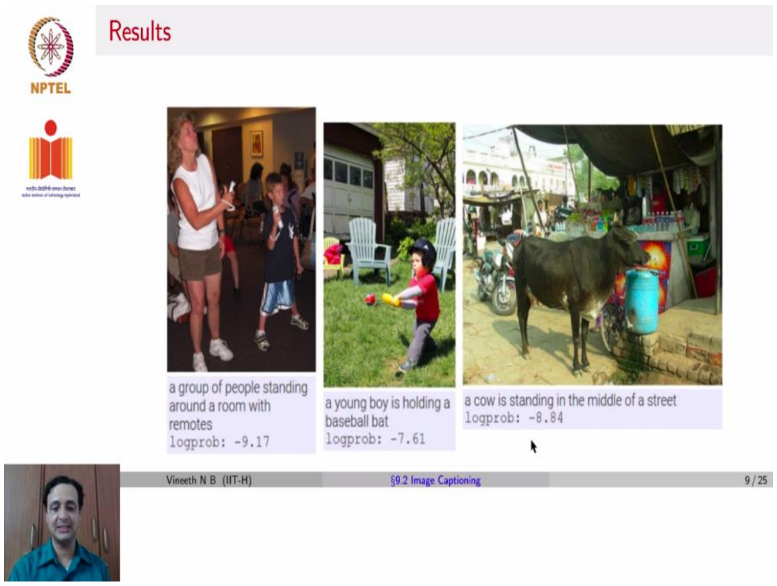
and that generates a new word. And if that final word is the end token, you know, you have completed your generation of words, which in this case, is Straw hat. What does it mean to generate words? How does the RNN generate words, in practice in implementation, it does not generate words, you have a vocabulary, it could be a set of thousand words or even more. The RNN predicts a softmax over your vocabulary.

And you pick the word with the highest probability, there are a few things that you can change when you pick that word from the vocabulary based on the softmax. But the most standard way of doing it is to pick the word which has the highest probability in your softmax vector. So in that sense, all of these are treated as classification problems. In this case, it is not image classes, but you have a vocabulary.

And the RNN predicts a softmax, over each word in your vocabulary. What is the start token then, if you had a thousand words in your vocabulary, the start token, and end token would be two additional words? So you would now have 1002 words. And when you give the start token, you would set a 1 at that particular location, and 0 everywhere else in 1002 dimensional vector. This is also known as a 1 Hot Vector.

Where the active word currently is given a 1 and every other word is set to 0. That is how such an RNN is implemented in practice. But coming back to captioning, given the image, you get an FC seven representation of basically you could also take a representation in an earlier layer that is given us input to the RNN and the RNN generates the phrase, Straw hat.

(Refer Slide Time: 16:58)



Results

a group of people standing around a room with remotes
logprob: -9.17

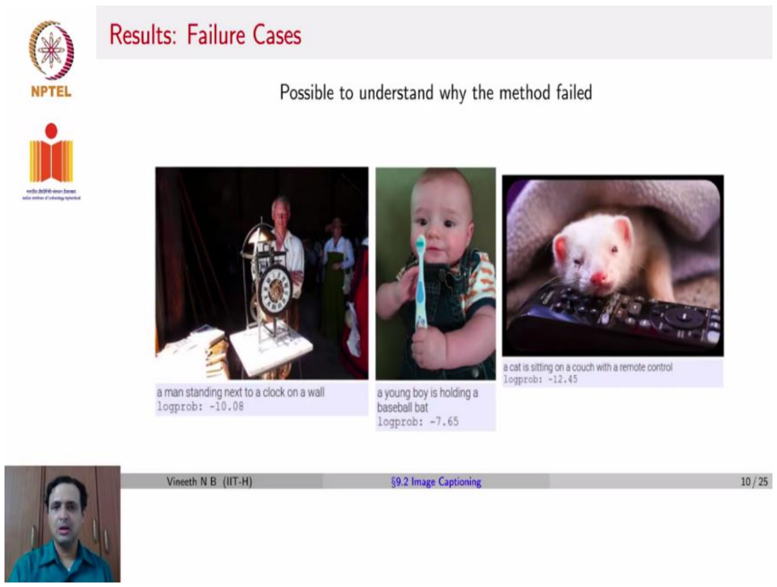
a young boy is holding a baseball bat
logprob: -7.61

a cow is standing in the middle of a street
logprob: -8.84

Vineeth N B (IIT-H) §9.2 Image Captioning 9 / 25

So here are a few results of such an approach. You can see here for this image, this approach predicts a group of people standing around a room with remotes, that is a fairly good caption, a young boy is holding a baseball bat. Again, a fairly good cat caption, a cow is standing in the middle of a street, fairly good caption again, the log probability, in a sense shows the loss that you get for this particular test instance.

(Refer Slide Time: 17:33)



Results: Failure Cases

Possible to understand why the method failed

a man standing next to a clock on a wall
logprob: -10.08

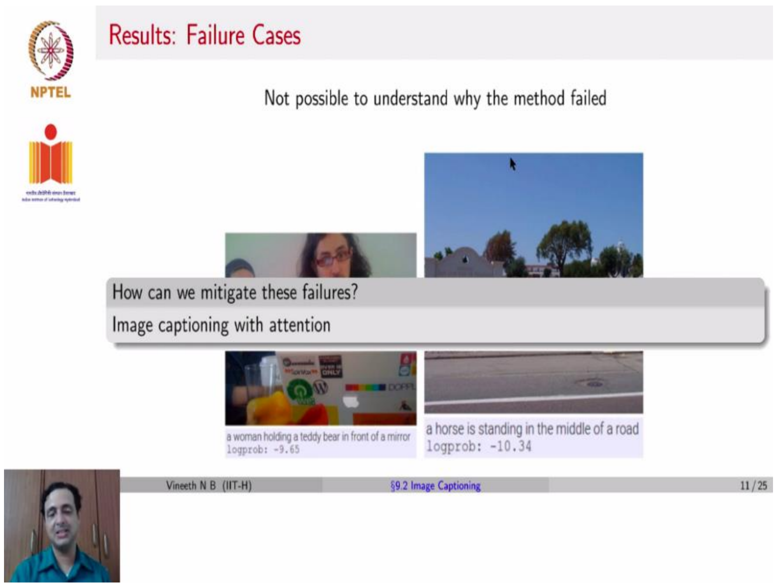
a young boy is holding a baseball bat
logprob: -7.65

a cat is sitting on a couch with a remote control
logprob: -12.45

Vineeth N B (IIT-H) §9.2 Image Captioning 10 / 25

On the other hand, here are some failure cases of such a model. In this particular case, for this image, the model predicts a man standing next to a clock on a wall. For this image, the model predicts a young boy is holding a baseball bat. And for this model, for this image, the model predicts a cat is sitting on a couch with remote control. All of these are wrong captions. But one can understand why the method failed. It was reasonably close to the output but was perhaps misled by a few elements and the interactions between the elements in the images. There are also other failure cases,

(Refer Slide Time: 18:18)


Results: Failure Cases

Not possible to understand why the method failed

How can we mitigate these failures?

Image captioning with attention

a woman holding a teddy bear in front of a mirror
logprob: -9.65

a horse is standing in the middle of a road
logprob: -10.34

Vineeth N B (IIT-H) §9.2 Image Captioning 11 / 25

which are more blatant. So you see here an image and the caption as a woman holding a teddy bear in front of a mirror. Or for this image, you have a horse standing in the middle of a road. For these cases, it is tough difficult to justify why the method failed. So how can we do better? How can we go beyond just CNN and RNNs to improve image captioning performance?

We already know the theme we already know the answer, we want to use Attention to help improve the performance. Let us now try to see in more detail how Attention could have been used to improve the performance with this kind of approach.

(Refer Slide Time: 19:08)

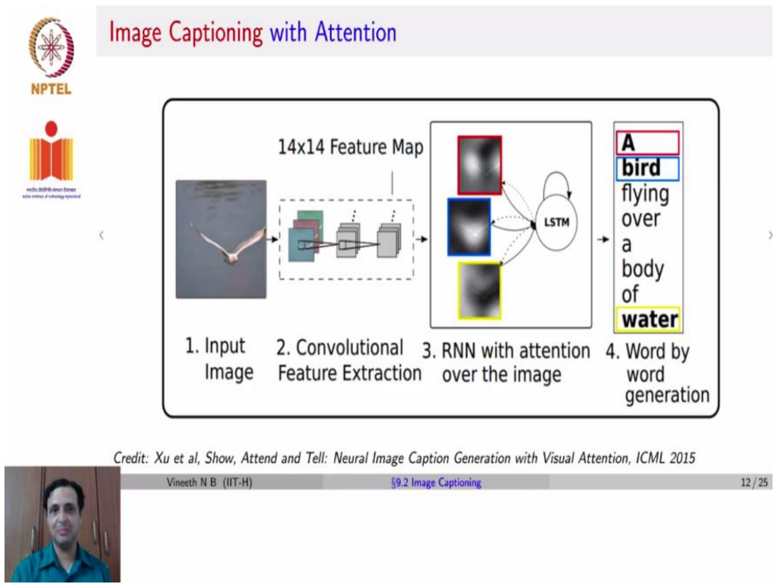


Image Captioning with Attention

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 12 / 25

So this method is based on a paper known as Show, Attend Tell, which was published in ICML in 2015. A very popular Image Captioning Approach. In this approach, you have an input image, you extract a certain feature map. Remember here, we mentioned this in the last lecture two. You do not consider FC seven layers, because if you need attention, you have to retain the spatial properties of the representation.

To extract a certain convolution layers output, in this case, say a 14 cross 14 feature map. Then you have an RNN with attention on this feature map and you finally get to word by word generation, a bird flying over a body of water. Let us now try to see how the RNN with attention is performed in this particular method.

(Refer Slide Time: 20:14)

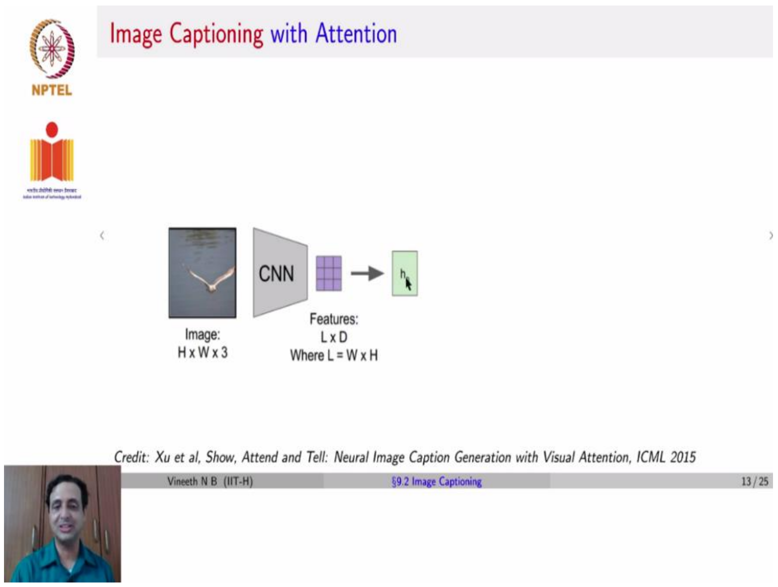


Image Captioning with Attention

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 13 / 25

Given an input image, whose dimensions are H cross W cross 3. You give that as input to a CNN. And the feature map after a particular convolution layer, let us assume is L cross D, where L is W cross H. Now, as we said, we give this as input to an RNN first hidden state H naught.

(Refer Slide Time: 20:42)

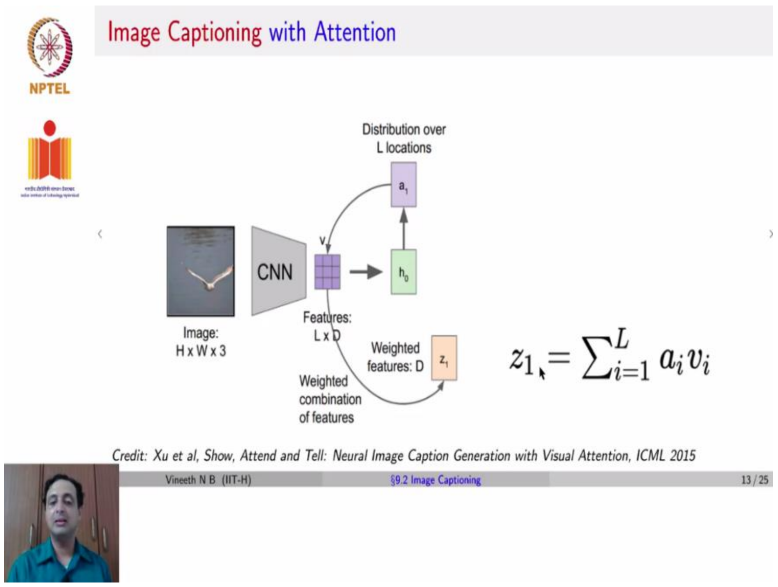


Image Captioning with Attention

$$z_1 = \sum_{i=1}^{L} a_i v_i$$

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 13 / 25

So H naught, in addition, to let us assume, say a start token, receives this feature map as input, an H naught gives this time distribution over locations. Recall, we just said that an RNN outputs a distribution over a vocabulary. This time, we are going to ask the RNN to generate a distribution over L possible locations.

What are these locations? These are the locations of each of these grids in the feature map. So we are going to ask the RNN, can you give me a softmax vector, which distributes attention over each of these grid locations? What do we do with this a1? a1 is a softmax vector, whose length would be the number of grid locations that you have. That could be H cross W, which we are going to denote as one single value L here.

Now we are going to use the a1, to wait, each of these locations in the feature map, and that becomes a weighted combination of features, which we denote as z1. z1 is given by ai vi. vi remember, are the features inside each of these grid locations that come as the output of the feature map. And that z1 now becomes the input to the RNN along with the first word, or the start token.

Now h1, which is the next hidden state of the next time step in the RNN gives us two outputs. It gives us a2 which is once again, a distribution over L locations, as well as d1, which is a distribution over vocabulary, which tells us what is the predicted word? What do we do with these two, a2 is once again multiplied with the feature map to give us a new weighted representation of the image feature map z2.

And d1, which is the word generated could be the input y2 or you could when you are training, you could just have the second word of the caption as input y2 to the next step of the RNN. So z2 and y2 are the inputs to the hidden, hidden state of the next step in the RNN. Once again, this generates a3 and d2, a3 is the distribution over L locations. d2 is the distribution or the vocabulary, which tells you what is the word a3 tells you which part of the image to focus on.

(Refer Slide Time: 23:52)

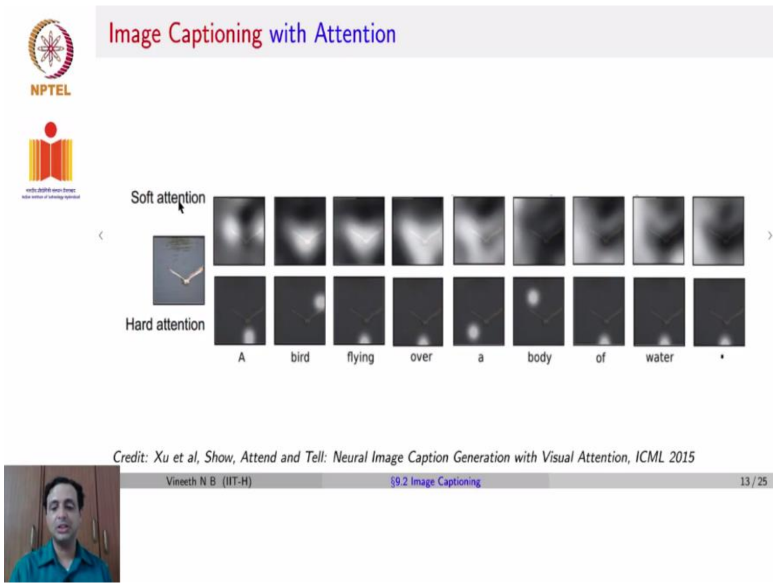


Image Captioning with Attention

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 13 / 25

This is repeated again and again. And the caption is generated. As we said in the previous lecture, there are two ways of going about this Attention model. Soft attention and Hard attention. In soft attention.

(Refer Slide Time: 24:13)

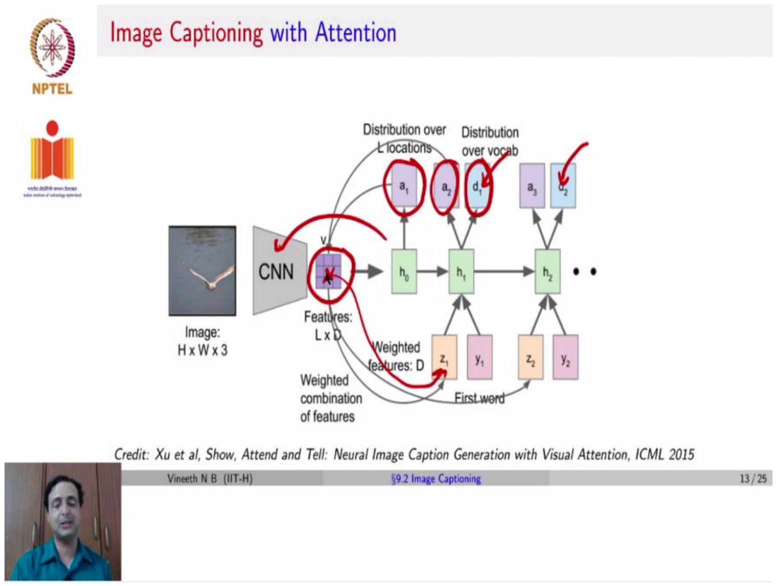


Image Captioning with Attention

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 13 / 25

The weight vector a1 is a softmax vector. And you use that as weights to multiply each of these grid locations in the feature map. So you have the z1 to be a1 into v, v1 depending on which entry you are looking at. A1 i into vi for instance. And this gives you soft attention or the feature map of the entire image. What is then Hard attention? Hard attention is when you look at that particular a1 softmax vector, pick one winning entry, and only use that grid location as the input to the next time step. This is known as hard attention.

Because you are not using a soft distribution of the weights across all parts of the image, you are going to choose a winning entry and pick only one part of the image will be one particular grid location here, maybe this location alone as input z1 to the next image. Are there advantages and disadvantages? Yes, indeed, hard attention requires an arg max at this particular step, because you want to consider the location which was the winning entry of the softmax vector.

And then our max is nondifferentiable so, you may not be able to use backpropagation to solve the heart attention approach. So, how is that implemented, it is implemented using reinforcement learning methods, which we will not focus on at this time. But the soft attention method at the same time is a weighted combination of all locations of the grid, which is deterministic and can be backpropagated through and hence can be learned through backpropagation.

So, how do you learn such a network using backpropagation? Remember, you have a distribution or locations, you have a distribution over vocabulary, the distribution or vocabulary, you know the word to be predicted. So you would have a cross-entropy error there. Similarly, for each step of the RNN, you would have a cross-entropy error. And using that, you can learn all the weights in the RNN.

And if required, you can also backpropagate that error back to the CNN to also fine-tune the CNNs weights. But the last year would be the sum of cross-entropy losses of each time step.

(Refer Slide Time: 27:08)

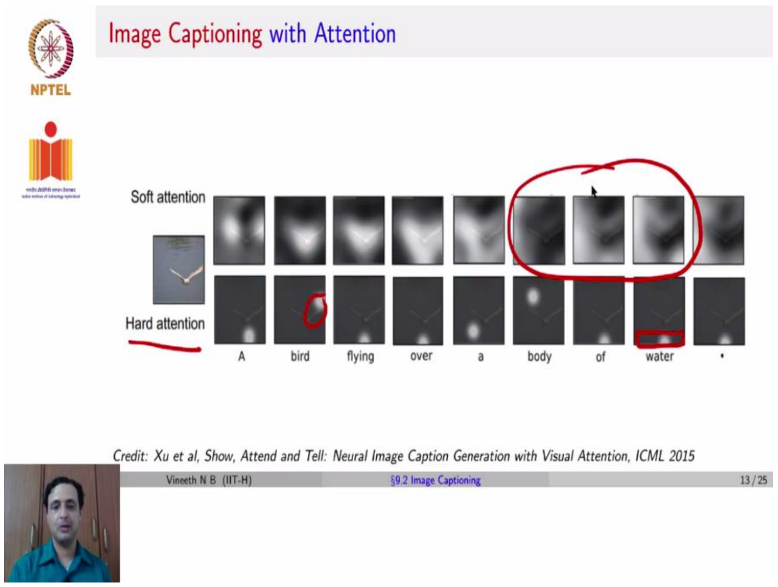


Image Captioning with Attention

Credit: Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015

Vineeth N B (IIT-H) §9.2 Image Captioning 13 / 25

Here is a visualization of Soft attention versus Hard attention. So you can see here that in soft attention, you focus your attention on different parts of the image. And when the method generates a bird flying over a body of water, you can see that initially, when it looks at a bird, that is where its attention is when it generates the word bird. That is where its attention is. And gradually, as it starts generating a body of water, it looks at everything other than the bird, if you see this part of it.

The bird is black, and the rest of the image is white, which means the attention was suggesting the model look at everything outside the bird to generate the words the body of water. In case of hard attention remember, the model focuses only on one patch. And when it comes to birds, it does focus on one part of the image which corresponds to the word, and it when it comes to the body of water. It does focus on one part of the image that corresponds to the body of the water.

But the localization may not be as evident, because we are choosing only one single part that is the winner of that softmax of a1 vector.

(Refer Slide Time: 28:33)

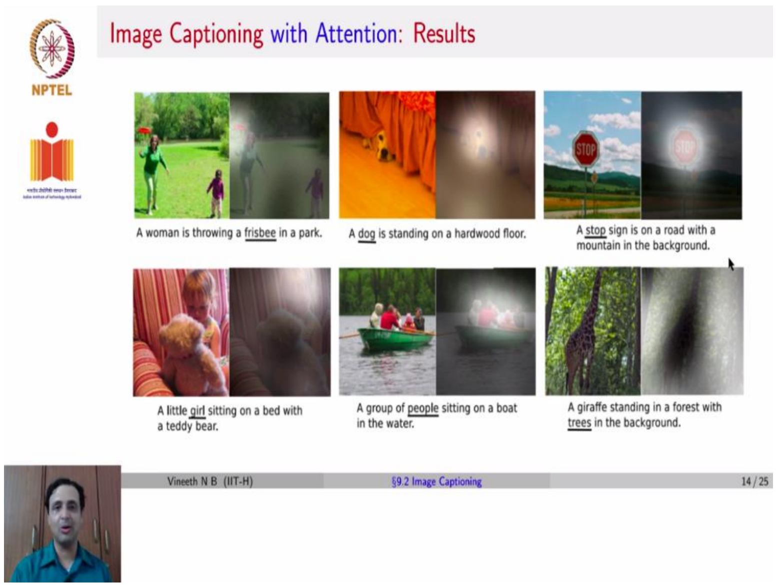


Image Captioning with Attention: Results

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

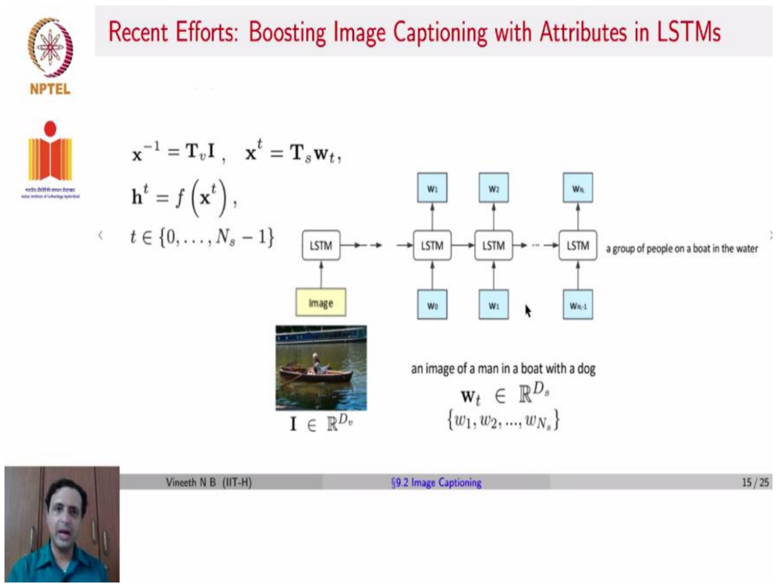
A giraffe standing in a forest with trees in the background.

Vineeth N B (IIT-H) §9.2 Image Captioning 14 / 25

Here are some results of this approach by using image captioning with attention based on this paper called, called Show Attend and Tell. So in this case, you can see that for this image, the caption generated by the model was a woman is throwing a Frisbee in a park. And you can see that when the word Frisbee was generated, the attention was completely on the Frisbee. Here are a few more examples; a dog is standing on a hardwood floor.

And when the model generated a dog, the attention was completely on the dog. A stop sign is on a road with a mountain in the background, the attention on stop sign when generated the word stop sign, and so on. The last image here is a challenging image where the generated caption says giraffe standing in a forest with trees in the background. And when it generated the word trees, it looked at everything other than the giraffe, which substantiates the choice in this particular context.

(Refer Slide Time: 29:44)



Recent Efforts: Boosting Image Captioning with Attributes in LSTMs

$$\mathbf{x}^{-1} = \mathbf{T}_v\mathbf{I}, \quad \mathbf{x}^{t} = \mathbf{T}_s\mathbf{w}_t,$$

$$\mathbf{h}^{t} = f\left(\mathbf{x}^{t}\right),$$

$$t \in \{0, \ldots, N_s - 1\}$$

an image of a man in a boat with a dog

$\mathbf{I} \in \mathbb{R}^{D_v}$

$\mathbf{w}_t \in \mathbb{R}^{D_s}$

$\{w_1, w_2, ..., w_{N_s}\}$

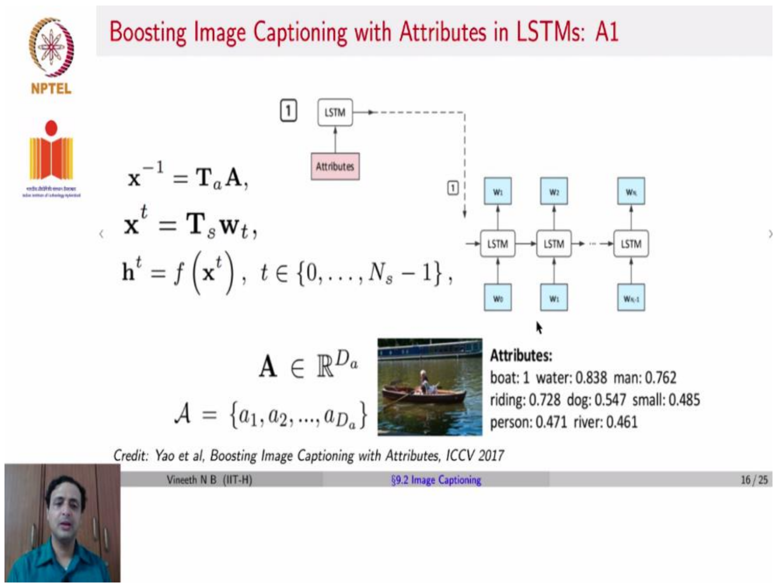
Vineeth N B (IIT-H) §9.2 Image Captioning 15 / 25

In terms of more recent efforts, so Show Attend and Tell was a paper published in 2015. More recent efforts have explored image captioning from a few different ways and let us see a couple of different examples. In one work in 2017, the idea was to see if the performance of image captioning could be boosted with attributes in LSTMs. So if you had an image I, and a set of words that form the caption, W.

So x at minus 1 is some transformation applied on the image by transformation Tv. That could just be a set of weights. And xt, which is the input at every time step is a transformation of each word in the caption. So this is the Vanilla Image Captioning approach that we have seen so far, where the image is processed, and then given to the first step of the LSTM. And each word of the caption is given as input at each time step of the unrolled LSTM.

And ht, which is the hidden state processes using some function f, the input at each time step. This is the vanilla version of using an image representation and the caption words to train the model. So this particular work tried a few different variants.

(Refer Slide Time: 31:22)



Boosting Image Captioning with Attributes in LSTMs: A1

$$\mathbf{x}^{-1} = \mathbf{T}_a \mathbf{A},$$
$$\mathbf{x}^{t} = \mathbf{T}_s \mathbf{w}_t,$$
$$\mathbf{h}^{t} = f\left(\mathbf{x}^{t}\right), \; t \in \{0, \ldots, N_s - 1\},$$

$$\mathbf{A} \in \mathbb{R}^{D_a}$$
$$\mathcal{A} = \{a_1, a_2, ..., a_{D_a}\}$$

Attributes:
boat: 1 water: 0.838 man: 0.762
riding: 0.728 dog: 0.547 small: 0.485
person: 0.471 river: 0.461

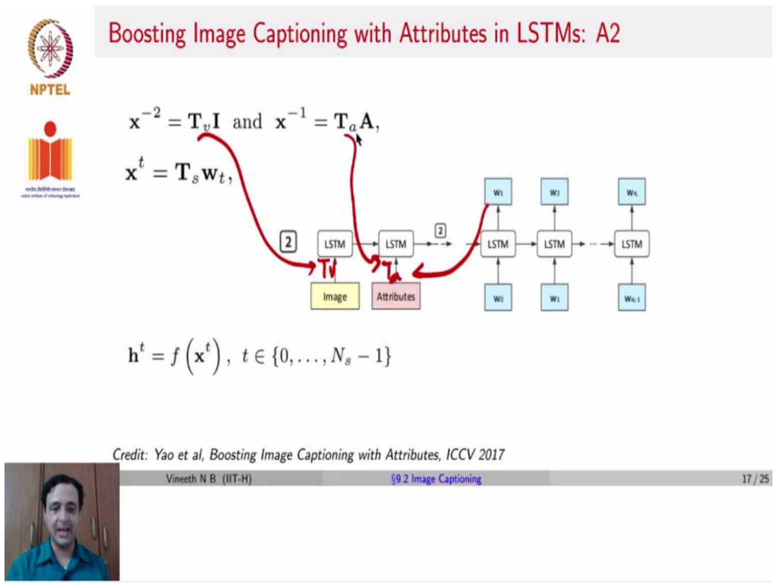
Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 16 / 25

Let us see each of these variants in one of these variants, which is the first one which we denote as A1, they considered transforming the images into attributes. So given an image, you could use an image classifier or you could use attributes that may be provided as metadata with the image. The attributes associated with this image are boat is 1, water, man, riding, dog, small, person and river with different probabilities.

Now, these become a set of attributes, an attribute vector, which comes from a set of attributes. And in this approach, the x at minus 1, which is the input that you give to the first hidden state, was the attributes and not the image. So the image is not given as input to your captioning model. It is only these attributes that you get out of this image. Why are we doing this? Because anyway, the output is a caption. So you may as well give a textual version of an image as input, the rest of it stays the same. This was A1 a variant.

(Refer Slide Time: 32:44)



Boosting Image Captioning with Attributes in LSTMs: A2

$$\mathbf{x}^{-2} = \mathbf{T}_v\mathbf{I} \text{ and } \mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A},$$

$$\mathbf{x}^{t} = \mathbf{T}_s\mathbf{w}_t,$$

$$\mathbf{h}^t = f\left(\mathbf{x}^t\right),\ t \in \{0, \ldots, N_s - 1\}$$

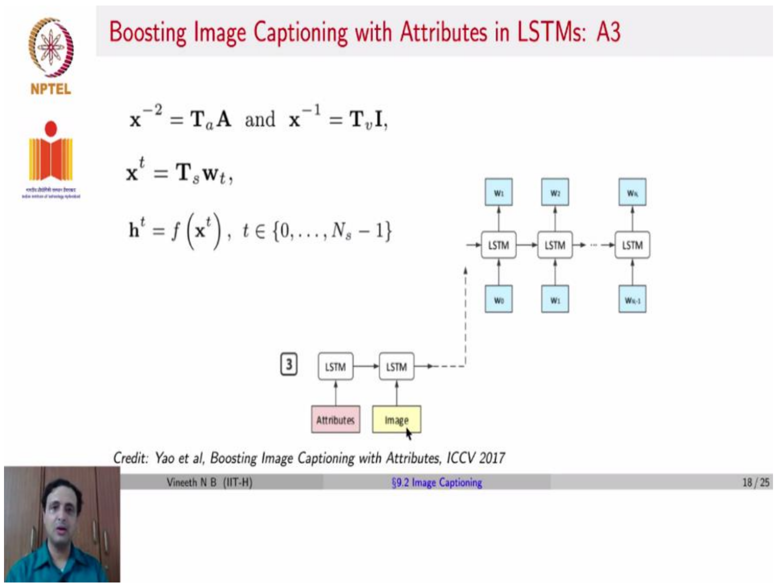
Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 17 / 25

In a second variant, A2 both the image and the attributes were given as input to the image captioning model. So the image is first given to the one-step of the LSTM of the RNN and in the subsequent time step, the attributes are given to the LSTM. So you have an x minus 2, which is based on the image, and an x minus 1, which is based on the attributes. Once again, Tv and TA are transformations that are given by learned weights.

So you could imagine that Tv would be a set of weights here. And TA would be a set of weights here, which are learned when you backpropagate from the output. Those are other weights that you learn. So this was a variant known as A2.

(Refer Slide Time: 33:37)



Boosting Image Captioning with Attributes in LSTMs: A3

$$\mathbf{x}^{-2} = \mathbf{T}_a\mathbf{A} \text{ and } \mathbf{x}^{-1} = \mathbf{T}_v\mathbf{I},$$

$$\mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t,$$

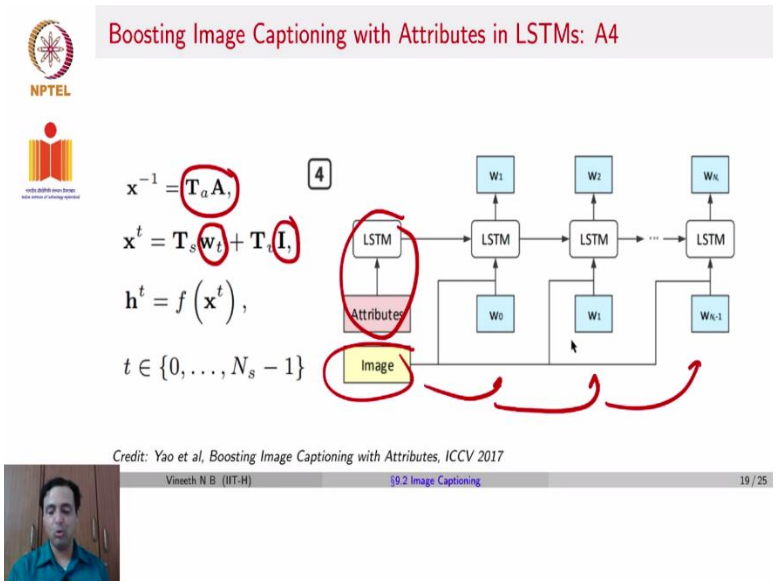
$$\mathbf{h}^t = f\left(\mathbf{x}^t\right), \ t \in \{0, \ldots, N_s - 1\}$$

Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 18 / 25

A3 was a variant, where the attributes were given first, and the image was given next, it is a variant of A2, where the inputs are swapped first attributes then the image.

(Refer Slide Time: 33:51)



Boosting Image Captioning with Attributes in LSTMs: A4

$$\mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A},$$

$$\mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t + \mathbf{T}_v\mathbf{I},$$

$$\mathbf{h}^t = f\left(\mathbf{x}^t\right),$$

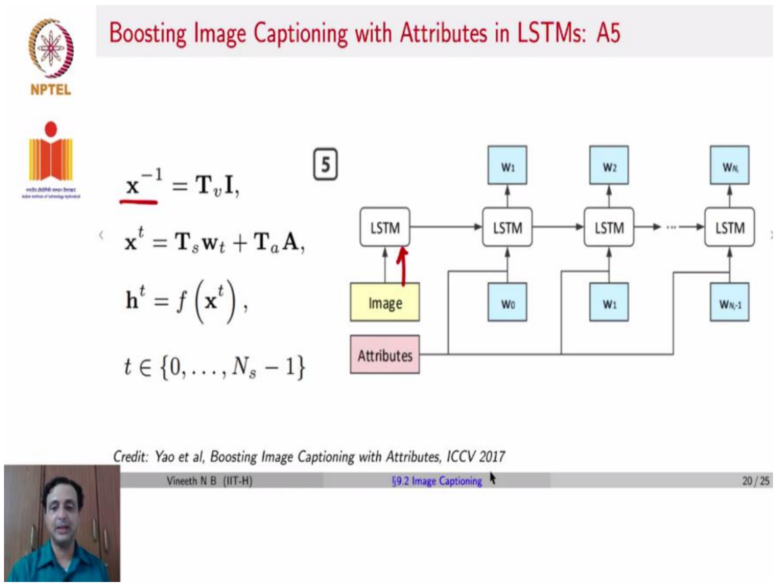
$$t \in \{0, \ldots, N_s - 1\}$$

Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 19 / 25

Then came a variant A4 were where at x minus 1, only the attributes were given, but at each time step, so at each time step, you give both the words in the caption and the image as input at each time step. So the attributes initially and the image at each time step, this was variant A4.

(Refer Slide Time: 34:21)



Boosting Image Captioning with Attributes in LSTMs: A5

$$\mathbf{x}^{-1} = \mathbf{T}_v \mathbf{I},$$

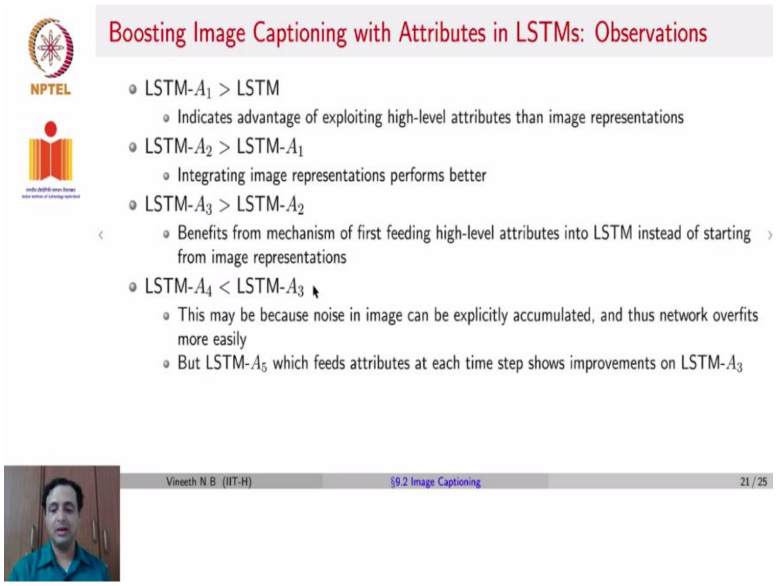
$$\mathbf{x}^t = \mathbf{T}_s \mathbf{w}_t + \mathbf{T}_a \mathbf{A},$$

$$\mathbf{h}^t = f\left(\mathbf{x}^t\right),$$

$$t \in \{0, \ldots, N_s - 1\}$$

Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 20 / 25



Boosting Image Captioning with Attributes in LSTMs: Observations

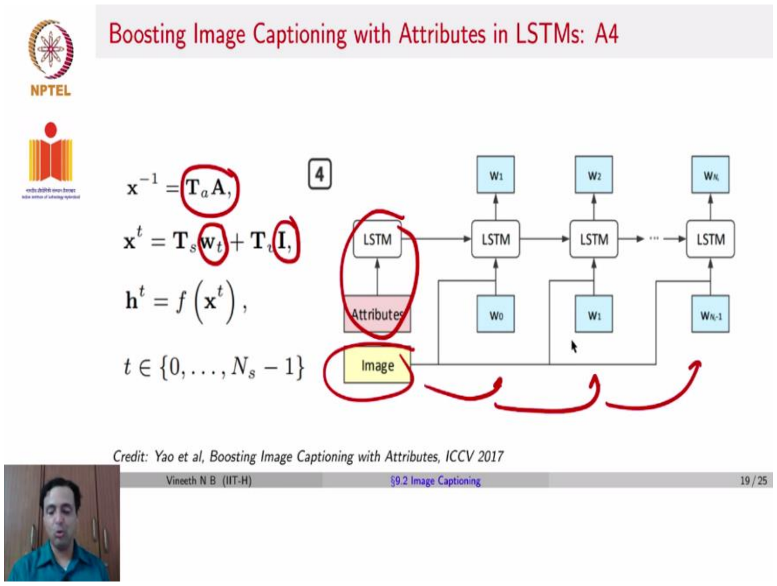
- LSTM-A_1 > LSTM
  - Indicates advantage of exploiting high-level attributes than image representations
- LSTM-A_2 > LSTM-A_1
  - Integrating image representations performs better
- LSTM-A_3 > LSTM-A_2
  - Benefits from mechanism of first feeding high-level attributes into LSTM instead of starting from image representations
- LSTM-A_4 < LSTM-A_3
  - This may be because noise in image can be explicitly accumulated, and thus network overfits more easily
  - But LSTM-A_5 which feeds attributes at each time step shows improvements on LSTM-A_3

Vineeth N B (IIT-H) §9.2 Image Captioning 21 / 25

And A5 was a variant where the image was given as input at x minus 1 and the attribute was given along with the caption in each time step, this was variant A5. So this particular work, which was by Yao et al, published in ICCV, 2017 concluded that A1 the variant A1 was better than LSTM, this indicated the advantage of using high-level attributes instead of image representations.

Variant A2 where the image representations were also integrated with attributes helped further improve performance. And similarly, by feeding the attributes first and the image later, A3 performed better than A2, then came a surprise that A4 did not perform, as well as A3.

(Refer Slide Time: 35:30)



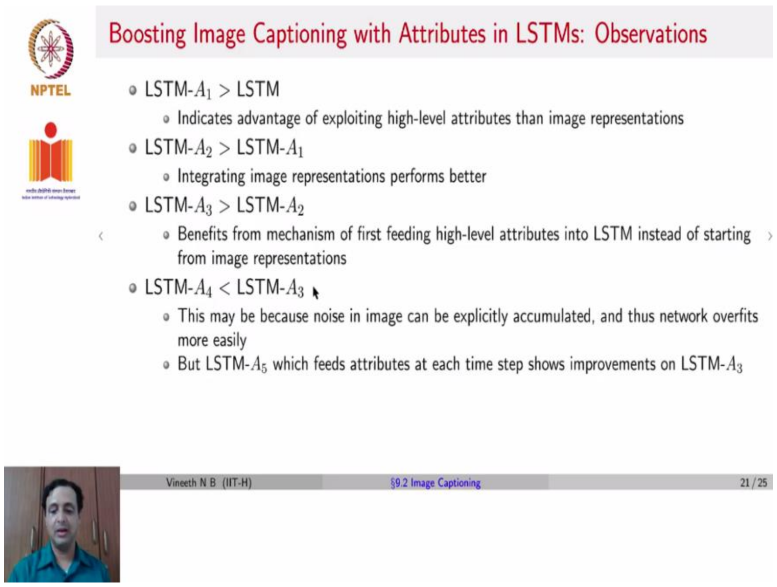
Boosting Image Captioning with Attributes in LSTMs: A4

$$\mathbf{x}^{-1} = \mathbf{T}_a\mathbf{A},$$
$$\mathbf{x}^{t} = \mathbf{T}_s\mathbf{w}_t + \mathbf{T}_i\mathbf{I},$$
$$\mathbf{h}^{t} = f\left(\mathbf{x}^{t}\right),$$
$$t \in \{0, \ldots, N_s - 1\}$$

Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Recall if you go back and see what A4 was, A4 was where you provided attributes at x minus 1, and the image as input at every time step.

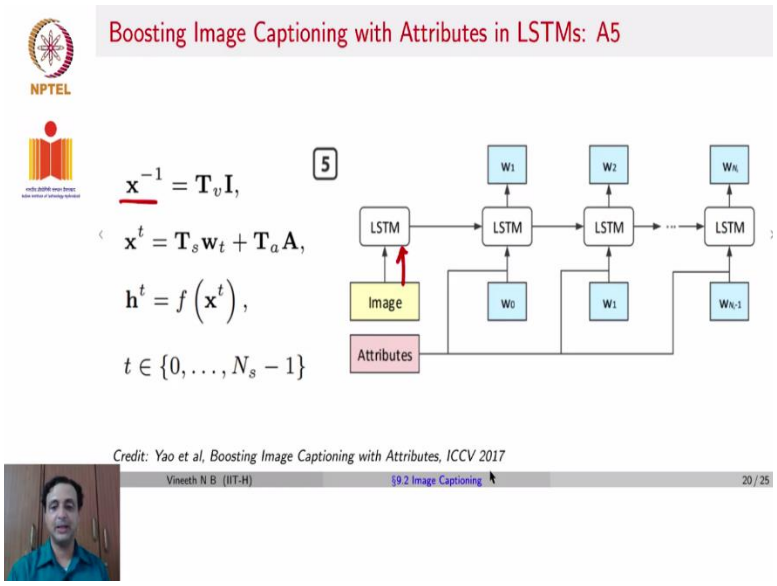
(Refer Slide Time: 35:42)



Boosting Image Captioning with Attributes in LSTMs: Observations

- LSTM-A_1 > LSTM
  - Indicates advantage of exploiting high-level attributes than image representations
- LSTM-A_2 > LSTM-A_1
  - Integrating image representations performs better
- LSTM-A_3 > LSTM-A_2
  - Benefits from mechanism of first feeding high-level attributes into LSTM instead of starting from image representations
- LSTM-A_4 < LSTM-A_3
  - This may be because noise in image can be explicitly accumulated, and thus network overfits more easily
  - But LSTM-A_5 which feeds attributes at each time step shows improvements on LSTM-A_3

Vineeth N B (IIT-H) §9.2 Image Captioning 21 / 25



Boosting Image Captioning with Attributes in LSTMs: A5

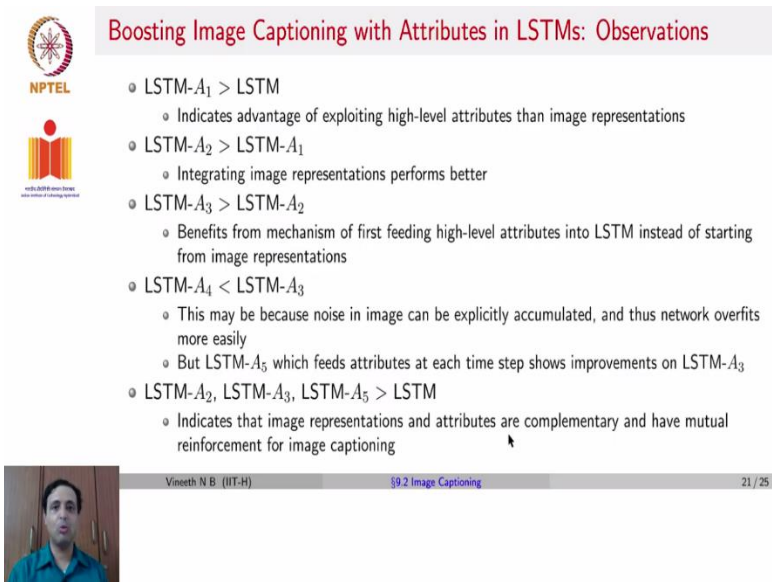
$$\mathbf{x}^{-1} = \mathbf{T}_v\mathbf{I},$$
$$\mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t + \mathbf{T}_a\mathbf{A},$$
$$\mathbf{h}^t = f\left(\mathbf{x}^t\right),$$
$$t \in \{0, \ldots, N_s - 1\}$$

Credit: Yao et al, Boosting Image Captioning with Attributes, ICCV 2017

Vineeth N B (IIT-H) §9.2 Image Captioning 20 / 25

Unfortunately, this did not perform as well. And a possible hypothesis for why it did not perform as well is that any noise in the image could now be accumulated in each time step of the RNN. And hence, the model could be suffering from some overfitting and not learning the relationship between the image and the text well. However, when this was swapped in A5 where you have the image given an x minus 1, and the attributes are given such at each time step.

(Refer Slide Time: 36:25)



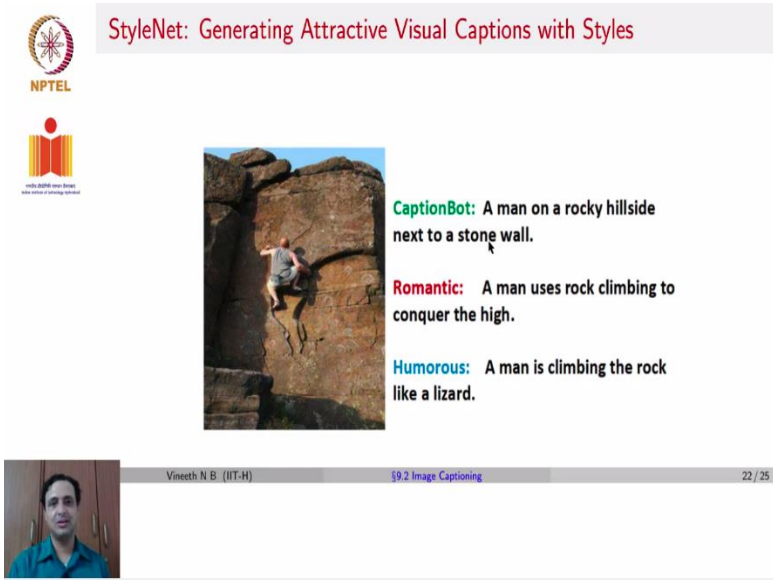
Boosting Image Captioning with Attributes in LSTMs: Observations

- LSTM-A_1 > LSTM
  - Indicates advantage of exploiting high-level attributes than image representations
- LSTM-A_2 > LSTM-A_1
  - Integrating image representations performs better
- LSTM-A_3 > LSTM-A_2
  - Benefits from mechanism of first feeding high-level attributes into LSTM instead of starting from image representations
- LSTM-A_4 < LSTM-A_3
  - This may be because noise in image can be explicitly accumulated, and thus network overfits more easily
  - But LSTM-A_5 which feeds attributes at each time step shows improvements on LSTM-A_3
- LSTM-A_2, LSTM-A_3, LSTM-A_5 > LSTM
  - Indicates that image representations and attributes are complementary and have mutual reinforcement for image captioning

Vineeth N B (IIT-H) §9.2 Image Captioning 21 / 25

Now, this model performed better than A3. And that gives the conclusion that image representations, as well as attributes, could be complementary. And both could help improve image captioning performance, depending on how you use them in the model.

(Refer Slide Time: 36:46)



StyleNet: Generating Attractive Visual Captions with Styles

CaptionBot: A man on a rocky hillside next to a stone wall.

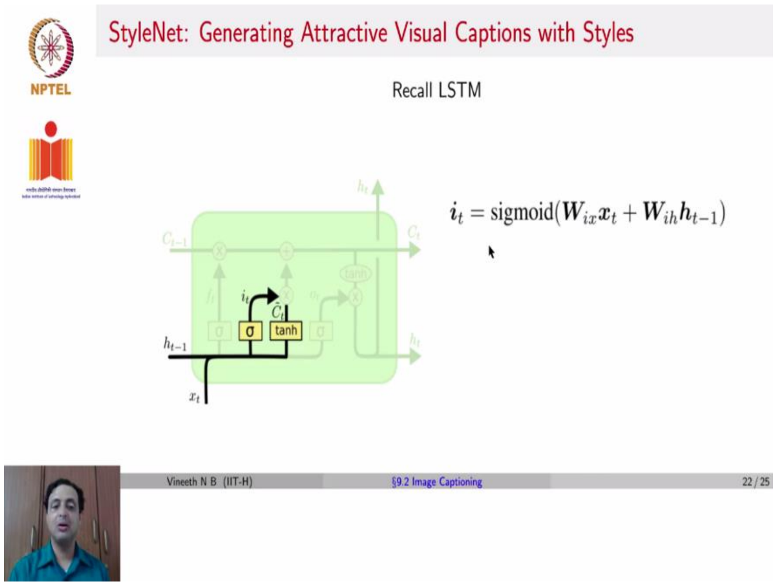
Romantic: A man uses rock climbing to conquer the high.

Humorous: A man is climbing the rock like a lizard.

Vineeth N B (IIT-H) §9.2 Image Captioning 22 / 25

Another interesting approach for image captioning was known as Style net, which tried to change the tone in the caption. So if you had an image, and the original caption was a man on a rocky hillside next to a stone wall, one could now give a romantic version of this, which says a man uses rock climbing to conquer the height. Or one could give a humorous version of this, which is a man is climbing the rock like a lizard? How do you change this tune in the caption?

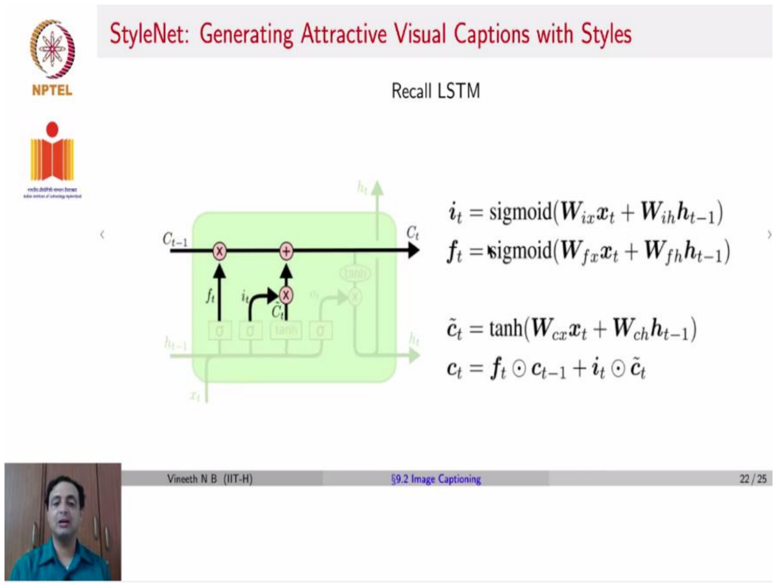
(Refer Slide Time: 37:30)



StyleNet: Generating Attractive Visual Captions with Styles

Recall LSTM

$i_t = \text{sigmoid}(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1})$

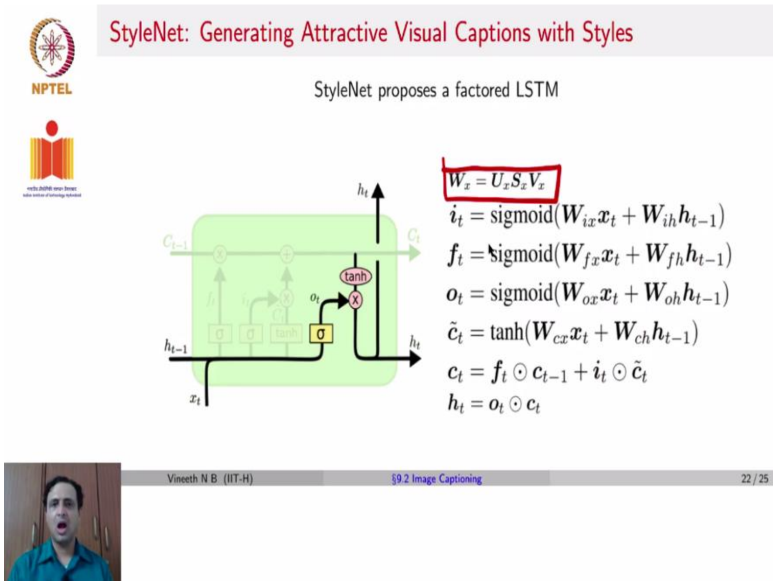


StyleNet: Generating Attractive Visual Captions with Styles

Recall LSTM

$i_t = \text{sigmoid}(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1})$
$f_t = \text{sigmoid}(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1})$

$\tilde{c}_t = \tanh(\boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1})$
$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$



StyleNet: Generating Attractive Visual Captions with Styles

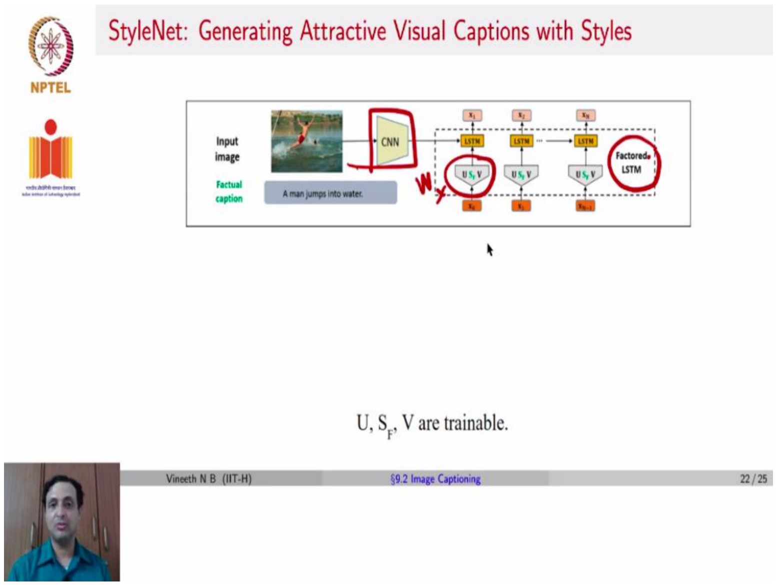
StyleNet proposes a factored LSTM

$$\boxed{W_x = U_x S_x V_x}$$
$$i_t = \text{sigmoid}(W_{ix}x_t + W_{ih}h_{t-1})$$
$$f_t = \text{sigmoid}(W_{fx}x_t + W_{fh}h_{t-1})$$
$$o_t = \text{sigmoid}(W_{ox}x_t + W_{oh}h_{t-1})$$
$$\tilde{c}_t = \tanh(W_{cx}x_t + W_{ch}h_{t-1})$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$h_t = o_t \odot c_t$$

Vineeth N B (IIT-H) §9.2 Image Captioning 22 / 25

To do this, recall our standard LSTM where we had our Input gate, forget gate, and output gate that was our LSTM. So this particular approach, suggests a change in the weight matrix of LSTMs, where it suggests that a weight matrix W xx especially can be written as Ux Sx Vx, this would be similar to an SVD decomposition. So they call this a factored SVM. How does this help?

(Refer Slide Time: 38:08)



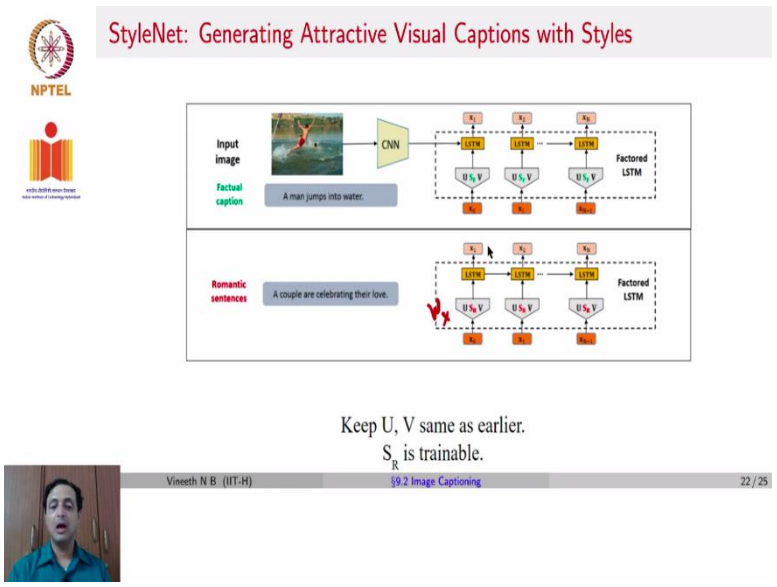
StyleNet: Generating Attractive Visual Captions with Styles

U, S_F, V are trainable.

Vineeth N B (IIT-H) §9.2 Image Captioning 22 / 25

Now, given an input image and a caption, you send the image through a CNN and then you use a factored SVM. now, where the weights Wx are now given by U Se Sf V for the factual caption.

(Refer Slide Time: 38:32)

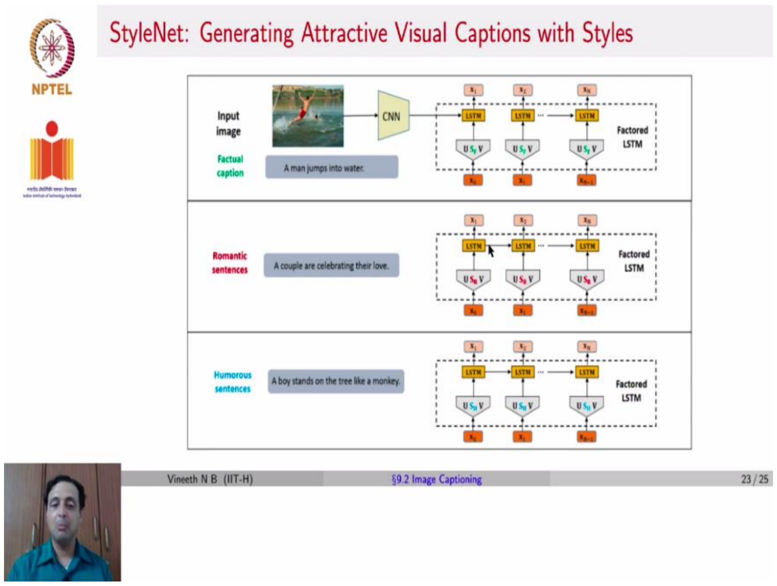


StyleNet: Generating Attractive Visual Captions with Styles

Keep U, V same as earlier.

S_R is trainable.

Vineeth N B (IIT-H) §9.2 Image Captioning 22 / 25

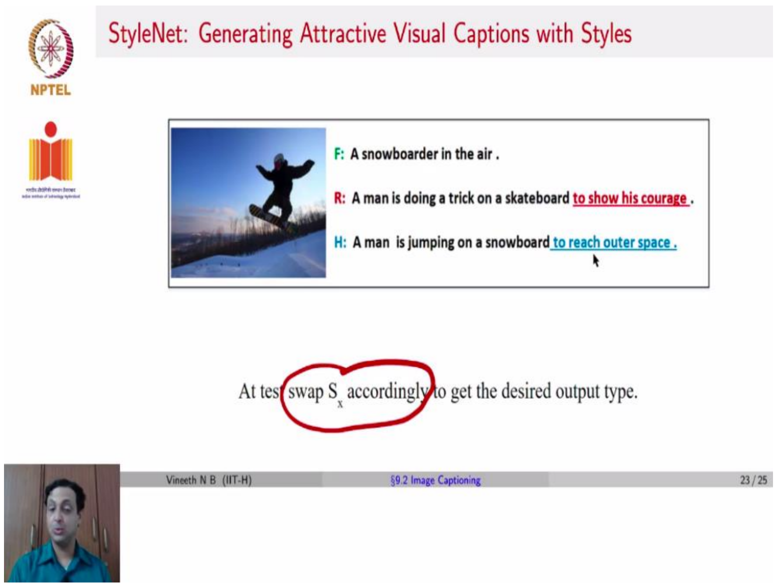


StyleNet: Generating Attractive Visual Captions with Styles

Vineeth N B (IIT-H) §9.2 Image Captioning 23 / 25

For the romantic caption, that way, Wx could be U Sr V and for the humorous caption, that will be given by U Sh V. Each of these matrices U, Sf Sr, Sh, and V are learned while training the model.

(Refer Slide Time: 38:56)



StyleNet: Generating Attractive Visual Captions with Styles

F: A snowboarder in the air .

R: A man is doing a trick on a skateboard to show his courage .

H: A man is jumping on a snowboard to reach outer space .

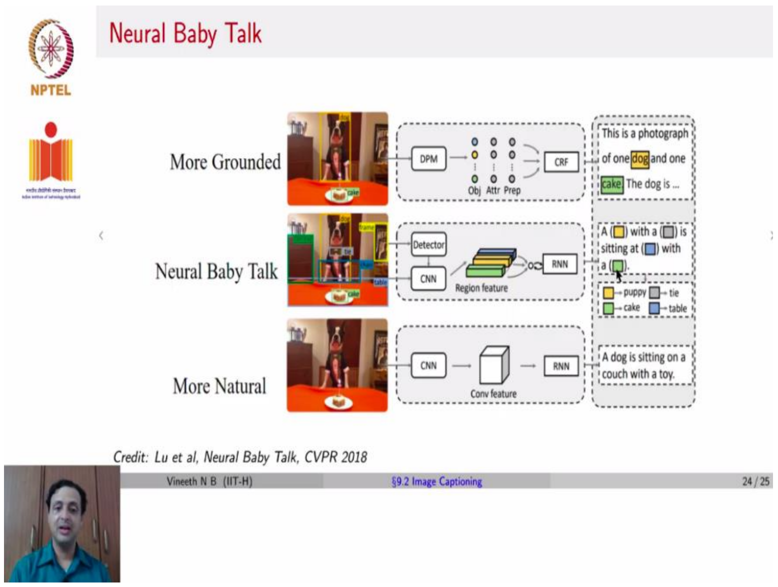
At test swap S_x accordingly to get the desired output type.

Vineeth N B (IIT-H) §9.2 Image Captioning 23 / 25

But now, at test time, we can swap the Sx accordingly to get the desired effect on the caption. So, a test time gave an image such as this, you could have a Factual caption with states a snowboarder in the air or a Romantic caption, which says a man is doing a trick on a skateboard to show his courage or a humorous caption, which is a man is jumping on a snowboard to reach outer space.

The tone may not be completely convincing, but it gives an idea of how you can vary what you need by factoring the S in the LSTM and then trying to change the S matrix in the middle to get different tones in the caption. This was Style net.

(Refer Slide Time: 39:55)



Neural Baby Talk

More Grounded
Neural Baby Talk
More Natural
DPM
Obj Attr Prep
CRF
This is a photograph of one dog and one cake. The dog is ...
Detector
CNN
Region feature
RNN
A ( ) with a ( ) is sitting at ( ) with a ( ).
puppy
tie
cake
table
Conv feature
A dog is sitting on a couch with a toy.

Credit: Lu et al, Neural Baby Talk, CVPR 2018

Vineeth N B (IIT-H) §9.2 Image Captioning 24 / 25

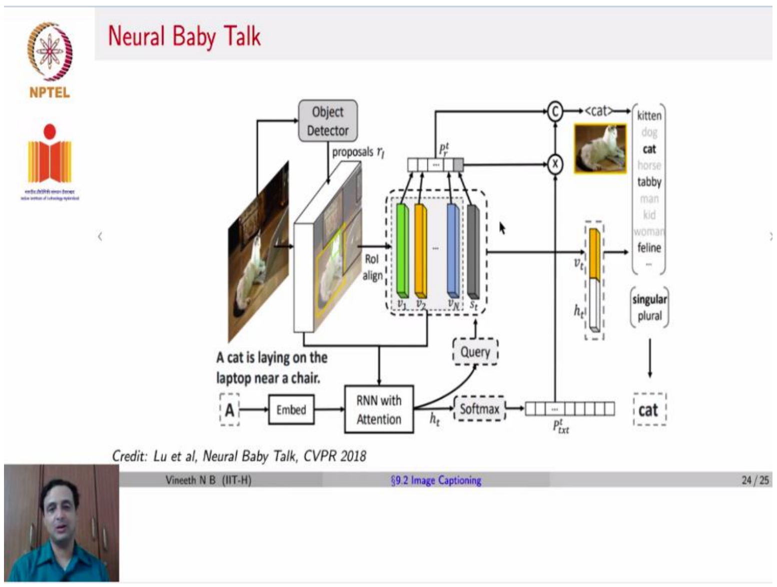
Another recent work was known as Neural Baby Talk, which is from CVPR of 2018, which again, provides a different spin on how image captioning is performed. So, in a more traditional computer vision sense before deep neural networks came. To be able to caption would have needed two specific kinds of approaches, known as Deformable Park Models, and Conditional Random Fields.

We will not get into these details now. But one could assume that using these methods, which are based on graphical models, you could say that this is a photograph of one dog and one cake. And maybe the caption continues after that, in a more natural sense, at this time, based on deep neural networks, given an image, you provide that to a CNN then an RNN, and you then have a dog is sitting on a couch with a toy.

Now can we go somewhere in between, that is what, this method tries to do, is neural baby talk says, Let us try to use detection models to be able to give better caption generation performance. So given an image, you have a detector, you get your region features. And you give these region features to an RNN to get your final generation, which states a box with a particular box is sitting at a box with a box where each of these boxes is different bounding boxes or region proposals.

The first yellow box is a puppy. The second one is a Tie. The green one is a cake. And the blue one is a table. So a puppy with a tie is sitting at a table with a cake. That is what would, would be done if you use deduction here.

(Refer Slide Time: 41:57)



Neural Baby Talk

Credit: Lu et al, Neural Baby Talk, CVPR 2018

Vineeth N B (IIT-H) §9.2 Image Captioning 24 / 25

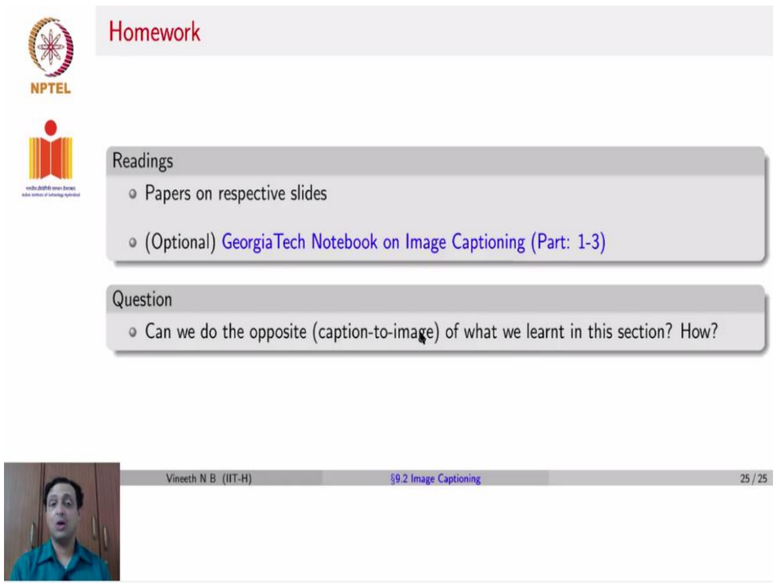
Let us see this model in more detail. So in the training phase, one would see that given an image using an object detector, such as a faster R-CNN model, or a mask, R-CNN model, you would get a set of region proposals on which if you use ROI align, which comes from mask R-CNN, you would get a set of aligned region proposals. Similarly, you also have a caption in the training face given in your training data set.

You transform that get a certain embedding of that caption. By embedding, we mean a representation of that caption, which is learned. How do we learn it? Using some weights, which are obtained through backpropagation? That is what we refer to as an embedding here; those embeddings are given to an RNN with attention along with each of these region proposals. So then you have two parts.

One, you have the bottom part here, which contains a softmax and gives you the textual output. That is the caption part of it, which you would use for inference or test time. The second part of it is using the RNN with attention, you decide which word in the caption that you are currently focusing on. And you take all these region proposals.

Together, you construct a vector, which is then given to a model, which predicts the label of that particular region in the image. So by combining the regions, and the textural prediction, one can predict the word and one can also say which part of the image was predicting the word.

(Refer Slide Time: 43:49)



Homework

Readings

- Papers on respective slides
- (Optional) GeorgiaTech Notebook on Image Captioning (Part: 1-3)

Question

- Can we do the opposite (caption-to-image) of what we learnt in this section? How?

Vineeth N B (IIT-H) §9.2 Image Captioning 25 / 25

So the homework for this lecture would be to read the papers on respective slides. It is all right if you did not understand every paper. This lecture was meant to give you an idea of different Image Captioning methods. And if you would like to get some hands-on, in addition to the assignment that we have, here is in the nice notebook on Image Captioning released by Georgia Tech.

The question for this lecture - Can we now do the opposite? Can we go from caption to image compared to what we learned in this lecture? How do we do this? Think about it and we will discuss the next time.