Reconstruction of hyperspectral images from RGB images

Pratish Pushparaj, Nishthavan Dahiya and Mayank Dabas

Department of Computer Science and Engineering, Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi, Delhi 110058, India

E-mail: pratish.pushparaj16@gmail.com, nishthavandahiya7@gmail.com, mayankdabas2401@gmail.com

Abstract. This paper accounts for the problem of construction of hyper-spectral (hs) images from RGB-images, i.e recovery of the whole spectral details/signature from a three-channel RGB image. The dataset used in this paper consists of ‘clean’ images, that are images without noise. There are 450 clean images along with correlative 450 hyperspectral images and all the pages are in PNG (.png) format. We approached this problem using 3 models Convhs_5, Enhanced-ResNet and Dense-HSCNN (D-HSCNN). These models increase in complexity from Convhs_5 to Dense-HSCNN. In the evaluation phase, Convhs_5 achieved average results with not so much clarity, whereas the best model came out to be Enhanced-ResNet with satisfying results, however, due to insufficiency of computational power/resources, we were not able to get the expected results from the Dense-HSCNN model. Whilst having more number of images in the dataset and more computational power can reduce the loss in all of the models furthermore.

Keywords: Deep learning, Convolution NN, Reconstruction, Hyperspectral images, RGB images

1. Introduction

In contrast to the traditional RGB imaging, the hyperspectral imaging captures information from across the electromagnetic spectrum instead of just RGB, including the spatial information of each pixel. Due to the spectral signatures of each pixel, detecting objects, light, and processes become much easier. Hyperspectral imaging systems have been in existence for a long, but their application in other than military services have been discovered recently. The primary use of hyperspectral images nowadays except for the military is in agriculture, astronomy, medical research, and surveillance.

In the earlier days, hyperspectral images were captured distinctly, that is, there were different techniques to capture hyperspectral images than capturing the simple RGB images explained in figure 1, namely:

- Spatial Scanning
- Spectral Scanning
- Non-Scanning
- Spatio-spectral Scanning

But the main disadvantage of these techniques was their very high cost and complexity. Very fast computers as well as sensitive detectors are required to process. And since the hyperspectral cubes are huge, multidimensional datasets, calculating total to a huge amount of data. So, storing such data and transferring it is quite difficult and not to mention costly so these techniques are quite inefficient.
Figure 1. Visual representation of hyperspectral imaging techniques used with spatial dimension (x, y) and spectral dimension (λ)

The idea of conversion of RGB images to Hyperspectral images came late into the light, but it was, and still is one of the most efficient ways to get a Hyperspectral image. Reconstruction of Hyperspectral image from an RGB image. The model-based methods for reconstruction are not enough as they cannot represent real-world spectral signatures, whereas learning-based algorithms overcome this disadvantage using an external dataset. But on the contrary, learning-based models will tend to overfit and will not be able to generalize, the reason for that being, it will probably do a brute-force mapping of the training data, moreover, the hyperspectral images formed in general are very different with varying centric wavelength and spectral signatures.

In this paper, we’re going to do a comparative study of different CNN-based methods for Hyperspectral Image reconstruction. We are going to use the NTIRE 2020 challenge dataset which consists of 450 spectral images for training. We will be comparing three types of models, a 5-layer basic CNN model, Enhanced-ResNet (which uses 10 layers), and a model proposed in the earlier NTIRE challenge called Dense-HSCNN.

In short, our contribution will be:
- Training the different models on the external dataset.
- Analysing the results and choosing the best model.

2. Related Works
In this section, we will go through the significant work that has been done earlier in this field of Hyperspectral imaging.

2.1. Hyperspectral imaging system
The basic hyperspectral imaging techniques have the usual trade-off between spatial and spectral resolution. Spatial scanning uses the slit spectra via projecting a strip onto a slit and dispersing with a grating. Spectral scanning, on the other hand, uses optical band-pass filters, exchanging them one by one, while the other platforms remain stationary. In the case of spatiotemporal scanning, a slit...
spectroscope is placed in front of a camera. Scanning is achieved when the whole setup is moved relative to the scene, or moving either the camera or the slit itself. A cheap way to convert RGB image to hyperspectral image is hyperspectral image reconstruction, but since a lot of information is lost in the process this method is not preferable. Through sparse coding or controlled illumination some progress has been made in this direction.

![Figure 2](image.png)

**Figure 2.** The basic function of the hyperspectral camera

### 2.2. Coded hyperspectral images (HSI) reconstruction method

Whilst older solutions depended on PCA for recapturing gamut from RGB images or any other multi gamut data, all of these methods were rapidly surmounted by methods which used sparse coding. Nonetheless, recently, the growth of hyperspectral images datasets has increased which allows us to think through neural network (NN) point of view, which gained popularity exponentially. At first less complicated NN were used like Radial Basis which was proposed by Nguyen. As the training data increased and became widely accessible, some very complex methods came into use. One such method was eighteen layered (GAN) which was suggested by Alvarez-Giza. Beside all these NN procedures, sparse coding is one of the very active approaches that is still followed. Robles-Kelly proposed a hybrid sparse coding/neural net approach, while Aeschbacher demonstrated that sparse coding approaches (i.e. adjusted anchored neighbourhood regression) can achieve comparable performance to those based on neural networks. In recent times the evaluation of performance on whole and sets of images are preferred over single spectra samples. The performance metrics vary widely between researchers as performance of hyperspectral reconstruction algorithms are more inclusive nowadays.

### 2.3. Construction of hyperspectral images using deep learning

Deep learning models like Pix2HS, CycleR GAN, HyperCNN, and CA-Net are used to convert RGB image to Hyperspectral image. These models are based on known techniques like CNN, GAN, and auto encoder models. Using these deep learning models will convert three channel RGB images to 31 channel Hyperspectral images.

#### 2.3.1. Pix2HS

This can be segregated into two parts, namely, the generator and the discriminators. The generator is based on the ResNet model and will be taking 512x512 RGB image as an input which will be converted to a hyperspectral image. It will output 31 channels images, where each channel will be representing a wavelength of spectral images. The discriminators, on the other hand, have got their roots in the Patch GAN model and it will take an RGB image and forecasted image to predict the originality of the Hyperspectral image by splitting them into the patches.

#### 2.3.2. CycleR GAN

This model has two generator models which are based on the U-Net model. The first generator will take an RGB image and the second generator will take a Hyperspectral image as its
input. Each generator has its associated discriminator models. The first discriminator will predict the originality of the original RGB image and image generated from the first generator. The second discriminator will predict the originality of the original RGB image and image generated from the second generator. Both the generators work together to reconstruct the image. This is sometimes referred to as the cycle consistency.

2.3.3. HyperCNN. CNN is an efficient and effective technique for fast reconstruction of the Hyperspectral images. It is a 5 layered model. The quantity of feature maps is restricted to 32 for the two layers in the beginning, whereas feature maps are restricted to 64 for the next two layers. The fifth layer will pop out 31 channel images. ReLU activation function is applied to each and every layer.

2.3.4. CA-Net. It is based on the AutoEncoder model. It recreates the output from the encoded input data CA-Net has two autoencoder based models. They are CA-Net 5 and CA-Net 10. CA-Net 5 comprises five convolutional and de-convolutional layers. CA-Net 10 extends the CA-Net 5 by having 10 convolutional and de-convolutional layers. Pooling is not used to prevent the data loss in the source image. Final output is going to be a Hyperspectral image with 31 channels.

3. Methodology
The GPU used to train the models is GTX 1050. We used Keras and TensorFlow framework, Adam optimizer with learning rate = 0.0001.

3.1. Dataset
The dataset was a collection of random 450 images of .png format. The resolution of each image was 512x482 and since they are RGB images they had 3 corresponding channels. Their pixel values are integers(uint8) between 0 to 255. Each RGB image has an equivalent hyperspectral image. They also had a resolution of 512x482 but had 31 different channels as they were hyperspectral. Their pixel values are float64 values between 0 and 1. Some examples of dataset images are shown in figure 3,4.
3.2. Data Preprocessing
The images were converted into a NumPy file the size of RGB images is 450x512x482x3 and the size of hyperspectral NumPy images is 450x512x482x31. For pre-processing, we normalize the RGB data. Then we resize the data to 450x256x256x(31 or 3) process explained in figure 5.

3.3. Convhs_5
Convhs_5 is a simple model with 5 convolution layers, and 7 convolution layers in the enhanced version. Padding size is kept the same in this, and then there is no pooling in the entire network, these steps are taken to keep every bit of information intact. This model is trained on the entire dataset of 450 images with 100 epochs with a batch size of 75. The model summary is explained in figure 6.

![Image](Convhs_5 model summary)

Figure 6. Convhs_5 model summary

3.4. Enhanced-ResNet

The Enhanced-ResNet model, as the name suggests uses Res-blocks, overall it is a 10 layer model, the number of Res-blocks used in this model is 3. Every Res-block has 2 convolutions and 1 ReLU activation. In this model too, padding is kept the same, also there is no pooling in this model too as we don’t want to lose any information from the images themselves. This model is trained on 150 images due to a lack of computational resources, with 100 epochs with a batch size of 30 images due to the complexity of the model. The model summary is explained in figure 7.

![Image](model_1_summary)

Figure 7. Model 1 summary
3.5. Dense-HSCNN
The Dense-HSCNN model is based on the HSCNN model presented in the 2018 NTIRE challenge, where many other models were presented as well. Initially, HSCNN was used as a method for hyperspectral recovery with an RGB image. First, HSCNN up-samples the input image through a spectral interpolation algorithm to obtain a hyperspectral image. However, Spectral Response function is required for up-sampling of the image. So only if the spectral response function is known, we can use HSCNN. This situation can be dealt with the use of densely-connected structure, known as Dense-HSCNN model. With the help of this dense structure we will reduce the vanishing gradient during the training.

![Figure 7. Enhanced-ResNet model summary](image)

![Figure 8. This is a configuration of hyper-params for HSCNN-D.](image)

This model also uses Res-blocks, although it is the most complicated model out of the three. The number of layers in this model is of the form (14+18b), where b is a variable. In simpler terms, the
HSCNN - D has a much deeper network and has higher reconstruction ability which makes it better than HSCNN and HSCNN - R. Moreover, the layers in this model are not all sequential, there are several parallel convolution layers as well, which overall improves the accuracy and efficiency of the model. If we have an RGB image(3 bands) as input, the result will be a hyperspectral image with 32 bands. We can see a decrease in the number of channels. Every dense block explicitly increases the number of channels which makes this an effective model for this problem.

4. Results
After training the model we tested it to predict the hyperspectral image of given RGB images and the results shown in figure 9,10,11. There was a significant difference seen between Convhs_5 and Enhanced-ResNet. Their predictions for the given test RGB images are shown in figure 12. As one can see the more accurate prediction of the two models is of Enhanced-ResNet's prediction image.

![Figure 9. Convhs_5](image_url)

![Figure 10. Enhanced_ResNet](image_url)

![Figure 11. Dense-HSCNN](image_url)
Figure 12. Ground Truth Images (RGB), Corresponding Convhs_5 prediction, Enhanced-ResNet predictions (from top to bottom)

5. Conclusion
In this paper, we compared various deep learning models for Hyperspectral Image reconstruction using RGB images. There was a significant difference between the Convhs_5 model and Enhanced-ResNet model as shown by the above test images. The value of loss function (Mean Absolute Error) was 0.0110
for enhanced Resnet and 0.0266 for Convhs_5. Convhs_5 with a very simplistic design has 5 convolution layers, on the other hand enhanced resnet has 10 layers and uses 3 ResBlocks which makes its prediction better.

Since we could only run for b=6 for Dense-HSCNN its loss came out to be about 0.0964 but the test image was very much similar to the test image produced by Enhanced-ResNet. Which means that if run the model with proper computational powers the Dense-HSCNN can give a better result than enhanced resnet. Since Dense-HSCNN is a much dense model it can only be run on computers with much better GPU or can be run on a supercomputer.

Table 1. Training and validation loss for models

| Model       | Training Loss | Validation Loss |
|-------------|---------------|-----------------|
| Convhs_5    | 0.0266        | 0.0254          |
| Enhanced-ResNet | 0.0110    | 0.0119          |
| Dense-HSCNN  | 0.0964        | 0.0908          |

6. Future Works
Numerous different training, testing experiments have been excluded from our research for the future due to lack of computational resources/power, time (i.e the training with full dataset was taking a lot of time in the case of Enhanced-ResNet and Dense-HSCNN, consuming even days for completing one pass through the dataset) and smaller dataset given that computational resources was not an issue. So for the future work, the things that need to be carried out are:

- To add the Attention model in all three models to enhance their accuracy.
- To train the Enhanced-ResNet model with the complete dataset and get the accuracy.
- To train the Dense-HSCNN model with suitable computing resources and with the complete dataset.
- To increase the dataset more than 450 images as the models tend to perform better with big dataset according to our results.

References

[1] Abdelhamed, Afifi M, Timofte R, Brown M 2020 NTIRE challenge on real image denoising: Dataset, methods and results Conf. on Computer Vision and Pattern Recognition Workshops (IEEE)

[2] Ancuti C O, Ancuti C, Vasluianu F A, Timofte R 2020 NTIRE challenge on nonhomogeneous dehazing Conf. on Computer Vision and Pattern Recognition Workshops, et al

[3] Arad B and Lin Y T 2020 challenge on spectral reconstruction from rgb image

[4] Aitken A P, Ledig C, Theis L, Caballero J, Wang Z, and Shi W 2017 Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize URL arxiv.org/abs/1707.02937

[5] Aeschnacher J, Wu J and Timofte R 2017 In defense of shallow learned spectral reconstruction from rgb images Proc. Int. Conf. Computer Vision Workshops (IEEE) pp 471-79

[6] Arad B and BenShahar O 2016 Sparse recovery of hyperspectral signal from natural rgb images Conf. on Computer Vision (Europe) pp 19-34
[7] Arad B, BenShahar O, Timofte R 2018 *NTIRE challenge on spectral reconstruction from rgb images Conf. on Computer Vision and Pattern Recognition Workshops* (IEEE), et al pp 1-4 p 10

[8] Arad B, Timofte R, Lin Y T, Finlayson G and BenShahar O 2020 *NTIRE challenge on spectral reconstruction from an rgb image Conf. on Computer Vision and Pattern Recognition Workshops* (IEEE), et al

[9] Dell’Endice F 2009 *Improving radiometry of imaging spectrometers by using programmable spectral regions of interest* (ISPRS J Photogramm Remote Sens) vol 64 issue 6 pp 632-39

[10] Shi Z, Chen C, Xiong Z, Liu D and Wu F 2018 *HSCNN+ Advanced cnn-based hyperspectral recovery from rgb images Conf. on Computer Vision and Pattern Recognition Workshops*

[11] Agahian F, Amirshahi S A and Amirshahi S H 2018 *Reconstruction of reflectance spectra using weighted principal component analysis Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Franc ais de la Couleur*

[12] Bhattacharya A, Karmakar R, Hans S, Govardhan M N 2016 *Conversion of RGB image to Hyperspectral using Deep learning URL analytics-vidhya/conversion-of-rgb-images-to-hyperspectral-using-deep-learning-f3dff0a1d60a*