

SuperResolution

José Pablo Ortiz

May 4, 2022

1 Introduction

In this work there is concise presentation of convolutional neural networks(CNN) and the architecture UNet. The purpose is to introduce all the basic concepts of convolutional neural networks and to show an application of this neural network for image super resolution.

2 Signals

A **bidimensional discrete signal** is an infinite matrix with entries x_{ij} , where x_{ij} is the activation of the position (i, j) [1]. The entries x_{ij} have a spatial relationship according to their position (i, j) . An example of bidimensional discrete signal is a grey scale digital image where $x_{ij} = 1$ means that the pixel (i, j) is black; while $x_{ij} = 0$ means that the pixel (i, j) is white, and any number between 0 and 1 is a grey tone.

A bidimensional discrete signal is called finite support if there exists K such that for every $|i| > K$ and $|j| > K$ $x_{ij} = 0$; then, this kind of signal can be represented as a finite matrix. In this work, we will only focus on finite supports signals. in order to explain the theory behind convolutional neural networks we need to make some definitions.

Definition 1. Let X be a matrix and $S, P \subseteq \mathbb{N}$ sets of contiguous natural numbers. The submatrix with rows S and columns p will be denoted by $X_{S,P}$ and will be called contiguous submatrix.

We are trying to create an expansion of X which contains some or all of the contiguous submatrices of X of dimension m_1 and m_2 . For this end we define the following sets

$$S_i = \{1 + is - 1, \dots, m_1 + is - 1\} \quad P_j = \{1 + pj - 1, \dots, m_2 + pj - 1\}. \quad (1)$$

The numbers $s, p \in \mathbb{N}$ are called height and width stride. They represen the separation that exists between the contiguous submatrices. On the other hand, in order for the contiguous submatrix to be well defined the values of i and j need to be:

$$i = 1, \dots, \frac{n_1 - m_1 + 1}{s} \quad j = 1, \dots, \frac{n_2 - m_2 + 1}{p}.$$

Definition 2. The contiguous submatrix expansion of dimension m_1 and m_2 with strides s and p of the matrix X is given by the following block matrix

$$\text{c.s.e.}(X; m_1, m_2, s, p) = \begin{pmatrix} A_{S_1, J_1} & \dots & A_{S_1, J_{k_2}} \\ \vdots & \ddots & \\ A_{S_{k_1}, P_1} & \dots & A_{S_{k_1}, P_{k_2}} \end{pmatrix}, \quad (2)$$

where S_i and P_j are defined as in (1) with $k_1 = \frac{n_1-m_1+1}{s}$ and $k_2 = \frac{n_2-m_2+1}{p}$.

This expansion can be seen as block matrix with all the rolling windows of dimension m_1 and m_2 with stride s and p . Once we have the expansion in (2) is quite easy to give an outline of how convolution neural networks work.

3 Convolutional operator

Definition 3. The convolutional operator \circledast between a signal X and $W \in \mathbb{R}^{(m_1 \times m_2)}$ with stride (s, p) is the matrix $Z = X \circledast W$ where the entries are given by

$$\begin{aligned} z_{ij} &= \sum_{u=s_i}^{s_i+m_1} \sum_{v=p_j}^{p_j+m_2} a_{uv} w_{uv} \\ &= 1_{m_1}^T (X_{S_i P_j} \odot W) 1_{m_2} \end{aligned}$$

where $X_{S_i P_j}^{(k)}$ is the contiguous submatrix of c.s.e. $(X; m_1, m_2, s, p)$ and $s_i = \min(S_i)$ and $p_j = \min(P_j)$.

Let X be an image in grey scale, within the context of convolutional neural networks, W is called filter or kernel because after convolving X and W ; W has the capacity to obtain features from X . Thus, the resulting matrix $Z = X \circledast W$ it is called **feature map**. Figure 1 is an example of this; where a kernel W was applied to X to obtain the outlines of the image.



(a) A subfigure



(b) A subfigure

Figure 1: A figure with two subfigures

3.1 Multiple channel convolution

In a more broad context and image $X \in \mathbb{R}^{(n_1 \times n_2) \times c}$ is a tensor where c is the number of channels in the image. The number of channels represents the number of characteristics that we have about X ; and the entry x_{ijk} the value of the k -th characteristic at the position (i, j) . For example, in a RGB image $X \in \mathbb{R}^{(n_1 \times n_2) \times 3}$ and every entry $x_{ij} = [r_{ij1}, g_{ij2}, b_{ij3}]$, is a combination of the different intensity of the primary colors (red, green and blue). To ease the notation the matrix $X[:, :, k]$ will denote the feature map of the k characteristic or channel.

Definition 4. The convolutional operator \circledast between $X \in \mathbb{R}^{(n_1 \times n_2) \times c}$ a multiple channel signal

and $W \in \mathbb{R}^{(m_1 \times m_2) \times c}$ with stride (s, p) is the matrix $Z = X \circledast W$ with entries

$$\begin{aligned} z_{ij} &= \sum_{k=1}^c \sum_{u=s_i}^{s_i+m_1} \sum_{v=p_j}^{p_j+m_2} x_{uvk} w_{uvk} \\ &= \sum_{k=1}^c 1_{m_1}^T (X[:, :, k]_{S_i P_j} \odot W^{(k)}) 1_{m_2} \end{aligned}$$

where $X[:, :, k]_{S_i P_j}$ is the contiguous submatrix of $e.s.c.(X[:, :, k]; m_1, m_2, s, p)$, $s_i = \min(S_i)$ and $p_j = \min(P_j)$.

In words, it's the sum of the convolution between each channel of X and W . This type of convolution is useful because the filter W moves around the spatial positions of X .

4 Resampling

Resampling is the process of changing the sampling rate or sampling frequency of a discrete signal to obtain a new discrete representation of the underlying continuous signal[2]. In this project we will use two types of resampling methods: DownSampling and UpSampling. When working with images DownSampling is the process of reducing the size of the image and UpSampling is the process of increasing the size of the image. There are many DownSampling and UpSampling techniques; nonetheless, in this project for DownSampling we will use max pooling and for UpSampling nearest-neighbor interpolation.

Definition 5. Let $X \in \mathbb{R}^{n_1 \times n_2}$ maxPooling operator of dimension m_1 and m_2 and stride (s, p)

$$z_{ij} = \max(A_{S_i P_j}), \quad (3)$$

where $A_{S_i P_j}$ is the block matrix of $c.s.e.(X; m_1, m_2, s, p)$.

Then if $Z = \maxPool(A; m_1, m_2, s, p)$ is a matrix of size k_1 and k_2 . When X is a multiple channel signal then the maxPooling operator is applied individually to every channel.

5 Layers

The layers that almost all CNN architectures use are convolutional, maxPooling and dropout. Nonetheless, in this application we will also use an UpSampling image.

5.1 Convolution Layer

The input of this layers is a feature map $X \in \mathbb{R}^{(n_1 \times n_2) \times c_0}$ and the output is another feature map $Z \in \mathbb{R}^{(n_1 \times n_2) \times c_1}$. In this layer we apply multiple channel convolutions between X and c_1 kernels of size (m_1, m_2) with both the stride width and height equal to 1. Each of this kernels consists of the weights/parameters that need to be estimated. Let $W \in \mathbb{R}^{[(n_1 \times n_2) \times c_0] \times c_1}$ be the c_1 kernels that need to be estimated then we define $\hat{Z} \in \mathbb{R}^{(n_1 \times n_2) \times c_1}$ as

$$\hat{Z}[:, :, k] = X \circledast W[:, :, :, k]. \quad (4)$$

The convolutional layer ends with the application of the ReLU activation function, that is:

$$z_{ijk} = \max(0, \hat{z}_{ijk}). \quad (5)$$

The reason a ReLU activation function is used is to add non linearity and to make the model more sparse by removing weak characteristics. This layer is the only one that has weights/parameters that need to be estimated.

5.2 MaxPooling Layer

The maxPooling layer has as input a feature map $X \in \mathbb{R}^{(n_1 \times n_2) \times c_0}$ and consists of applying maxPooling to each of the characteristics of X . That is, if $Z = \text{maxPooling}(X; m_1, m_2, s, p)$ then

$$z_{ijk} = \max(X[:, :, k]_{S_i P_j}) \quad (6)$$

and $Z \in \mathbb{R}^{(k_1 \times k_2) \times c_0}$.

5.3 UpSampling Layer

The UpSampling Layer has as input a feature map $X \in \mathbb{R}^{(n_1 \times n_2) \times c_0}$ and consists of applying the nearest-neighbor interpolation upSampling algorithm to upscale by 100% . Thus, the output will be $Z \in \mathbb{R}^{(2n_1 \times 2n_2) \times c_0}$

6 UNet

In this section we will present a method to UpSample an image from size 16×16 to size 32×32 . For this end, we will use the UNet. The UNet was introduced by in 2015. And it can be classified as convolutional Encoder-Decoder. 'Encoder-Decoder models are a family of models which learn to map data-points from an input domain to an output domain via a two-stage network: The encoder, represented by an encoding function $z = f(x)$, compresses the input into a latent-space representation; the decoder, $y = g(z)$, aims to predict the output from the latent space representation.'

[3] Both the Encoder and the Decoder are convolutional neural networks in blocks. A block is a combination of successive applications of the layers in section 5. The innovation of the UNet is that each block of the Decoder is connected to a block in the Encoder. The blocks used in the encoder will be called Down Block and the blocks used in the decoder will be called Up Block.

6.1 Down Block

Figure 2 shows the composition of a DownBlock. 'It consists of the repeated application of two 3x3 convolutions (same convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels.'

6.2 Up Block

Figure 2 shows the composition of a UpBlock. 'Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU.'

6.3 Tail

The tail consists of an UpSample layer followed by two 3x3 convolutions, each followed by a ReLU.

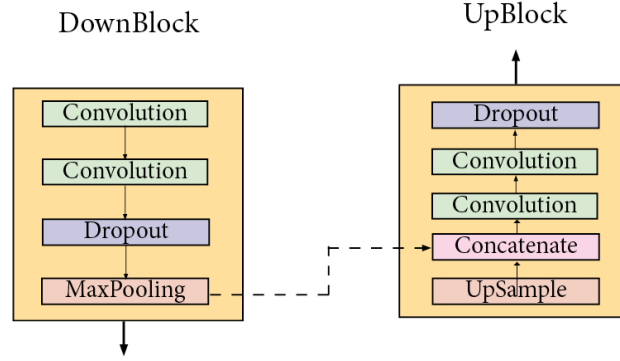


Figure 2: The DownBlock and UpBlock along with the connection they have.

6.4 Model

Figure 3 shows the architecture of the Unet used in order to up sample an image by 2x. Hence, the model is a function $f_\theta : \mathbb{R}^{(16 \times 16) \times 3} \rightarrow \mathbb{R}^{(32 \times 32) \times 3}$ where θ is all the matrices W given in a convolutional layer.

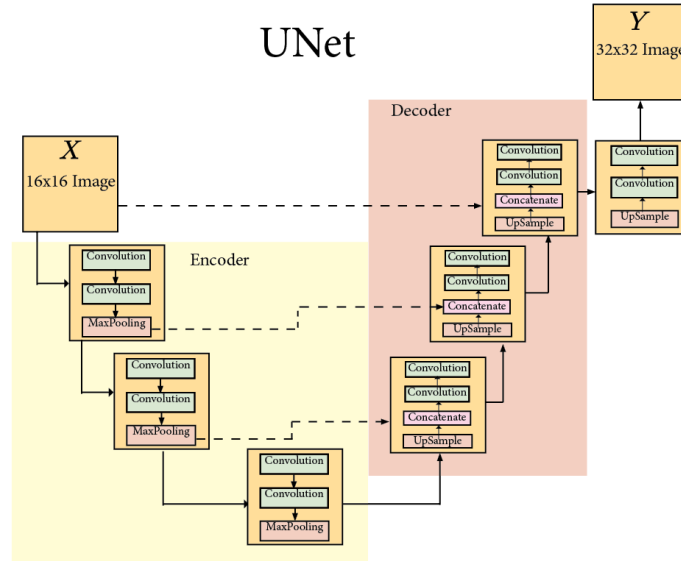


Figure 3: TheUNet: in yellow the encoder and in red the decoder.

7 Training

7.1 Dataset and preproces

The dataset was gather from Wikipedia and it consisted in images of famous paintings from different periods and places. Then we gather all the contiguous submatrix of size $(32, 32)$ with stride $(32, 32)$. Everyone of this images of size $(32, 32)$ was then downsample using nearest-neighbor interpolation to a size of $(16, 16)$.

The dataset will be have as inputs the images of size $(16, 16)$, $X \in \mathbb{R}^{(16 \times 16) \times 3}$, and outpus the corresponding images of size $(32, 32, 3)$, $Y \in \mathbb{R}^{(32 \times 32) \times 3}$. Thus, our data will have $n = 23300$ observations which was split with 90% training and 10% test.

7.2 Loss

Thus, given an observation (X, Y) the loss for a observation will be given by the Frobenius norm of the difference:

$$L(X, Y) = \sum_{(X, Y) \in \text{train}} \|f_{\theta}(X) - Y\|_F. \quad (7)$$

The algorithm that it's used to minimize (7) is minibatch stochastic gradient descent with the gradient calculated through backpropagation and being updated using the ADAM method.

8 Results

The results will be compared to an example given in wikipedia[5], where an image of painting by Evelyn De Morgan is upscaled using different software/techniques: PaintShop Pro, waifu2x and Topaz A.I. Gigapixel with Low noise reduction.

References

- [1] Calin, O. (2020). *Deep Learning Architectures: A Mathematical Approach* (1st ed.). Springer.
- [2] Oppenheim, A. V., & Schafer, R. W. (2013). *Discrete-Time Signal Processing: Pearson New International Edition*. Pearson Education.
- [3] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavazc, N., & Terzopoulos, D. (2020). *Image Segmentation Using Deep Learning: A Survey*.
- [4] Ronneberger O, Fischer P,& Brox T (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*.
- [5] Wikipedia contributors. (2022, February 4). Image scaling. Wikipedia. https://en.wikipedia.org/wiki/Image_scaling#Deep_convolutional_neural_networks.



Figure 4: Original

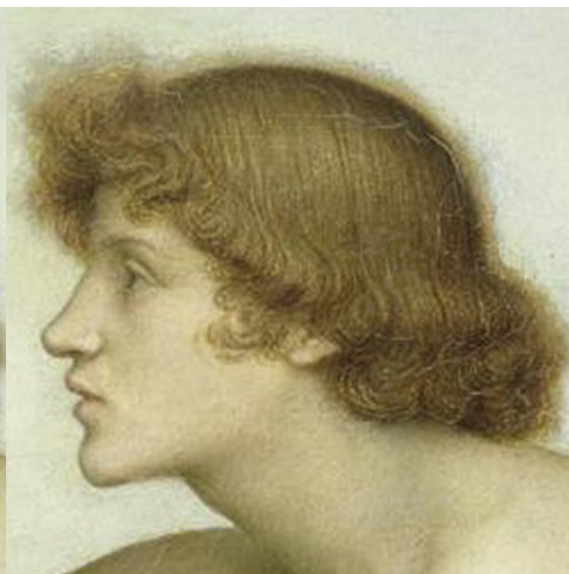


Figure 5: PaintShop Pro

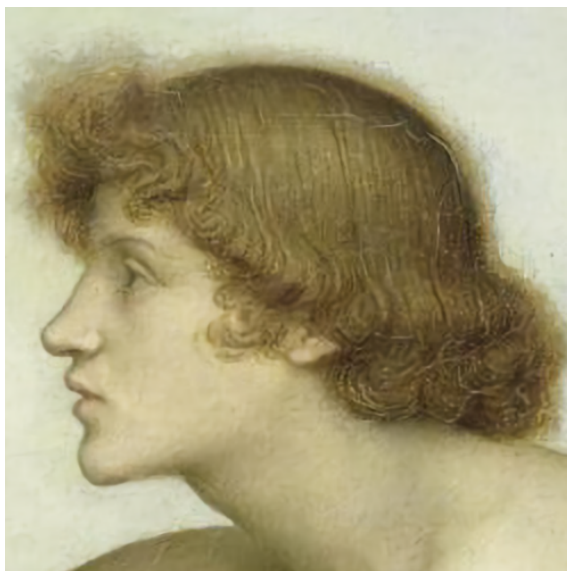


Figure 6: Waifu2x



Figure 7: UNet