LaMa network architecture search for image inpainting

Dmytro Kolodochka[1], Marina Polyakova[1], Oleksandr Nesteriuk[1] and Victor Makarichev[2]

[1] *Odesa Polytechnic National University, 1, Shevchenko Ave., Odesa, 65044, Ukraine*
[2] *National Aerospace University "Kharkiv Aviation Institute", 17, Chkalova Street, Kharkiv, 61000, Ukraine*

Abstract

The neural architecture search problem is to obtain a neural network architecture with a version of the selected block that has the best performance according to a pre-selected evaluation strategy compared to other alternative versions. The aim of the paper is to improve the performance of image inpainting using neural architecture search by applying the wavelet transform to the LaMa network. Analyzing the results of experiments on researching the performance of image inpainting using the developed software it was noticed that the inpainting was better for images containing significant areas of uniform intensity, fine-grained or structural texture. Fragments of images, including complex textures or detailed patterns were inpainted worse. The proposed technique for searching neural architecture for image inpainting based on LaMa differs in the ratio of image inpainting time and the quality of the reconstructed image. Inpainting of images with large masks based on the LaMa network is improved by applying the wavelet transform. In particular, the quality of filling the missing areas with image edges and small details is improved. In addition, it was researched the dependence of the quality of generating of details and edges of objects in the image on the properties of the image textures, which can be described by texture descriptors. Prospect for further research is prediction the effectiveness of the image inpainting with the LaMa networks depending on the estimated values of original image texture descriptors and missing areas size.

Keywords

Image inpainting, neural architecture search, wavelet transform, LaMa network

1. Introduction

In many applications in computer vision systems and computer graphics it is nessesary to fill missing areas of images. Image inpainting is applied in many practical situations, such as removing redundant elements or restoring damaged parts of photo [1]. Another application is the photo retouching of fabrics, skin and hair taking a lot of time when done manually. Image inpainting can also be applied to video, because video is a sequence of frames. Often, due to compression, some parts of the video can be damaged, and advanced image inpainting methods are able to solve this problem effectively. These methods are also useful for museums with limited budgets that cannot hire a professional artist to restore paintings.

The research object is natural image inpainting in computer graphics and computer vision systems.

Image inpainting methods are classified into two main types, specifically, direct methods and deep learning methods. Direct methods include methods based on partial differential equations, semi-automatic drawing, texture synthesis, for example, PatchMatch, implemented in Adobe Photoshop. Direct methods are fast, require almost no computing resources, easy to implement, and process images of any size. But the filling of missing areas by direct methods is based only on known areas of the same image. Therefore, it will be impossible to restore objects that have no analogues in the image. In addition, direct methods poorly restore large missing areas of images [2].

Deep learning image inpainting methods can generate missing areas of the image with fine local textures and good global consistency. Thus, DeepFill v1-2 [3, 4], EdgeConnect [5], CoModGAN [6] differ in properties of reconstructed images, processing time, the size of the processed image, and the quality of filling of image regions [7]. The shortcoming of the listed methods is the unsatisfactory results of generating both image context and texture when using large masks.

ICST-2024: Information Control Systems & Technologies, September 23-25, 2023, Odesa, Ukraine.
dmitrytdr@gmail.com (D. Kolodochka); marinapolyakova943@gmail.com (M. Polyakova); nesteryuk@op.edu.ua (O. Nesteriuk); v.makarichev@khai.edu (V. Makarichev)
0009-0006-3329-1504 (D. Kolodochka); 0000-0001-7229-7657 (M. Polyakova); 0000-0002-0806-8259 (O. Nesteriuk); 0000-0003-1481-9132 (V. Makarichev)

In general, to achieve better results in image inpainting, the convolutional neural network (CNN) architecture is complicated or divided into sub-networks with separate tasks [8]. LaMa-Fourier uses a fewer parameters and a single network instead [9]. It is able to obtain a good result even when the missing areas occupy most of the image.

The subject of the research is methods of image inpainting based on LaMa network.

The LaMa-Fourier network allows to fill large areas of spectral textures with high performance. But the LaMa-Fourier inpaints the fine details of images and edges of objects with insufficient quality. The filling missing areas of structural textures such as grass, leaves, textile is also difficult for LaMa-Fourier network. To avoid this shortcoming, the LaMa-Wavelet processes both global and local features of images using a wavelet transform [7]. However, there are quite a lot of options for using the wavelet transform in CNN architectures. Neural architecture search allows to design the effective LaMa network architecture with the lowest loss on the training set of images.

The aim of the paper is to improve the performance of image inpainting with neural architecture search by wavelet transform applying in the LaMa network. The neural architecture search is justified the selection of network to solve the image inpainting problem.

The main contributions of the paper are following.

The neural architecture search technique is elaborated for image inpainting.

The results of neural architecture search allow choosing a solution of the image inpainting problem based on the requirements to the quality of restored images and processing time.

It was revealed and researched the dependence of the quality of generating of details and edges of objects in the image on the properties of the image textures, which can be described by texture descriptors.

2. Problem statement

The RGB natural image is given by formula $I(x,y)=(I_R(x,y), I_G(x,y), I_B(x,y)\}$, where $x=1, ..., n$; $y=1, ..., m$; $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the color channels. To model image missing areas a mask is introduced. It is a binary image $M(x,y)$ of the same size as each channel of the original image. The mask is element-by-element producted by image channels, and the image with missing areas is defined as $I_M(x,y)=(I_R(x,y)\cdot M(x,y), I_G(x,y)\cdot M(x,y), I_B(x,y)\cdot M(x,y))$. The image $I_M(x,y)$ should be transformed so as to fill missing areas. In this case, the resulting image should be an approximation of the original one in the sense of some criterion [7, 9].

The problem of neural architecture search is as follows [10, 11]. The neural architecture search space is included basic network with alternative versions of selected block. Further the performance estimation strategy must be selected. It evaluates the performance of neural networks which architecture includes one of the versions of selected block. Finally, it is necessary to obtain the neural network architecture with version of selected block which has the best performance compared with other alternative versions.

Let a set F includes the architectures $struct_{f\theta}$ of the inpainting network $f_\theta(\bullet)$: $F=\{struct_{f\theta}\}$, where θ is network weights. F is noted as search space. Taking the image with missing areas $I_M(x,y)$ the CNN $f_\theta(\bullet)$ processes the input in a fully-convolutional manner, and produces the inpainted image $I_{in}(x,y) = f_\theta(I_M(x,y))$ which approximates the original image $I(x,y)$.

Then the dataset D consisting of pairs (image $I(x,y)$, mask $M(x,y)$) is selected to train network $f_\theta(\bullet)$ with architecture $struct_{f\theta} \in F$ by training strategy T. It is necessary to find an architecture $struct_{f\theta} \in F$ within a given time or computation budget t which has the lowest possible validation loss L_{val} when trained using dataset D and training strategy T:

$$\min_{struct f\theta \in F} L_{val}(f_\theta, \theta^*), \tag{1}$$

where $\theta^* = \arg\min_\theta L_{train}(f_\theta, \theta)$ [10], L_{train} denotes the training loss. The pairs (image $I(x,y)$, mask $M(x,y)$) of dataset D obtained from natural images and synthetically generated masks.

3. Literature review

In this paper a neural architecture search is performed to improve the quality of image inpainting. Then the wavelet CNN architectures are reviewed to generate network blocks for search space

forming. Analysis of the deep learning CNN architectures allowed the authors to determine the main applyings of wavelets in network architectures.

The first approach is wavelet embedding in CNN layers. Thus, in [12, 13] the wavelet transform is used as an implementation of convolutional and pooling layers to obtain a multiscale representation of the image.

In [14, 15] it is proposed to use a wavelet transform instead of a pooling layer in the CNN architecture for image recognition to reduce the dimension of feature maps. The max pooling and average pooling are most popular pooling methods. They based on a neighborhood processing that easily introduce visual distortion. To avoid this problem, a pooling method based on Haar wavelet transform was elaborated [14]. In [15] more sophisticated Daubechies wavelet and coiflets are used to perform the pooling. Additionally, a new pooling method for CNNs is proposed combining multiple wavelet transforms. The benefit of these pooling methods is to improve the performance of object recognition.

Although, in neural networks a few activation functions have been applied, the new activation function search still being an open research area. In [16] Gaussian family wavelets (first, second and third derivatives) are reused as activation functions in neural networks. The combination of these activation functions improve the CNNs performance. In [17] to enhance the proposed wavelet CNN the activation function of the convolutional layers is replaced by a real part of Gabor wavelet, but the activation function of the last layer is sigmoid function. Thus, the precision and accuracy of image classification on the test datasets was improved.

The second approach is the alternation of wavelet transform levels with CNN layers [12, 18]. In [12] it is first noticed that CNNs process images directly in the spatial domain. To incorporate a spectral approach into CNNs, a multiresolution analysis and CNN are combined into one model via wavelet transform and integrated it as additional components in the network architecture. The applying of wavelet CNNs in texture classification and image annotation problems allows to achieve the better accuracy than existing CNNs while having significantly fewer parameters [12]. In [18] multi-level wavelet CNN architecture is designed to include the CNN block before each level of discrete wavelet transform (DWT). The CNN block is a fully convolutional network without pooling, inputting all wavelet coefficient subbands. Each layer of the CNN block is composed of convolutional layer, batch normalization (BN), and ReLU activation. The last layer of the multi-level wavelet CNN is a convolutional layer which is adopted to predict resulting image. The experimental results show the effectiveness of multi-level wavelet CNN with Haar and Daubechies wavelets for image restoration problems such as image denoising and removal of JPEG image artifacts.

The third approach is CNN with wavelet domain inputs. In [19] it is noticed that although pooling layers reduce the computation requirements to CNNs, they cause the loss of information and affect the image classification accuracy. The CNN with wavelet domain inputs is proposed to enhance the quality of input information and increase the classification accuracy without changing the overall structure of the pre-defined CNN or enlarging a number of parameters. Specifically, at the pre-processing stage wavelet packet transform or dual-tree complex wavelet transform is applied to original image. Then some wavelet coefficient subbands are selected as the CNN inputs so that the networks are directly trained in the wavelet domain. Experiment shows the improving of the image classification accuracy.

Analyzing the considered CNN architectures using the wavelet transform the following was noticed. The time for image processing by a trained network is most likely comparable for the three approaches considered, but the training time for a CNN with wavelet domain inputs is significantly less, since training examples can be pre-transformed into the space of wavelet coefficients before training the network.

Besides, the use of wavelets in convolutional layers is characterized by the complexity of implementation and interpretation of the results. In addition, this approach limits the image feature extraction the learning process, which was an advantage of CNN. The use of the wavelet transform in the pooling layer has shown effectiveness in object recognition, but such layers are not used in LaMa networks.

CNNs with wavelets as activation functions significantly increases the amount of computation compared with ReLU. In addition, it is difficult to predict how such an applying of the wavelet transform will affect the result of image inpainting.

Using a CNN with wavelet domain inputs limits the selection of features to the domain of wavelet coefficients. In addition, the quality of inpainted image is likely to be negatively affected by the border effect of the wavelet transform.

Interleaving of wavelet transform levels with CNN layers, we search image features emphasizing image details and object edges. Although the feature selection is somewhat limited, this approach is easier to implement. Therefore this and previous approach are used further in this paper to form a neural architecture search space.

4. Materials and methods

To improve the LaMa network performance, the technique of the neural architecture search for image inpainting is proposed. The stages of this technique are as follows.

1. Underlying neural architecture is selected based on the analysis of existing CNN for image inpainting.
2. To construct the neural architecture search space some basic network block is selected for modification. The alternative versions of selected block are generated and included in neural architecture search space.
3. The loss function is defined to estimate the training and validation losses.
4. Image dataset is selected and pairs (image, mask) is formed to train networks with designed architectures.
5. The measures of image inpainting performance are selected.
6. Each network with selected architecture is learned to fill missing image areas on training set of images. The validation set of images is used to control the learning process.
7. The trained networks are applied to test images and the image inpainting performance is evaluated depending on the size of missing areas.
8. The obtained results are analyzed to determine which neural architecture is better inpainted images from considered database depending on missing areas size.

If necessary, the elaborated technique of neural architecture search can be configured to research other CNNs which fill missing image areas, as well as to identify disadvantages and validate the obtained results. Further the implementation of the technique of neural architecture search for image inpainting is considered. The first, second and third stages are discussed in this section. The following sections examine the remaining stages.

At the first stage to generate a neural architecture search space for image inpainting, a base CNN architecture is selected [10, 11]. The architecture of the LaMa-Fourier network is shown in Figure 1 [9]. The network is inputted an image with pixels need to be inpainted. Further, this image is downscaled by a factor of 3 and processed by nine residual blocks. After that, the image is rescaled to its original size and fed to network output [9].

In the residual block, the double Fast Fourier Convolution (FFC) decomposes the image into local and global textures which are further passes through the convolution layers [9]. The global texture additionally processed by spectral transform block. Then the convolution layer outputs are added "cross over cross". BN and the ReLU activation are applied to them. The results of processing of global and local texture are concatenated and summed with the original image (Figure 2).

In the spectral transform block the real and imaginary parts of Fourier transformed image are concatenated (Figure 2). Then there are sequentially applied the convolutional layer, BN and the ReLU activation function. The obtained result is splitted on the real and imaginary parts which are prosessed by inverse fast Fourier transform (iFFT). The result of iFFT is the output of the block [9].

To implement the second stage of the proposed technique, notice, that neural architecture search is a very time consuming. Probably, that is why this approach has not been used to design CNNs for image inpainting. At least such papers have not been finding by authors of this paper. Because of much time training it is therefore unrealistic to use a large search space. To form the search space in this paper only the spectral transform block on Fig. 2 is considered. This block processes the global context of the image. It was designed the four versions of such block using DWT.

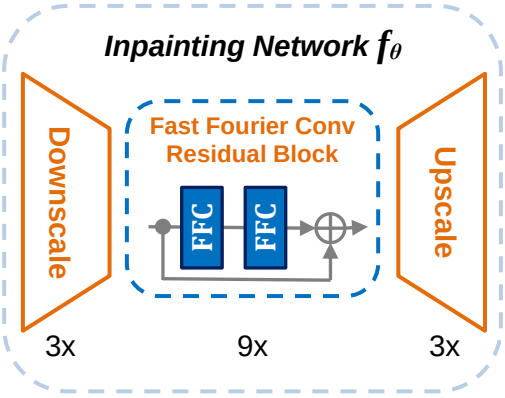


Figure 1: LaMa-Fourier network architecture [9]

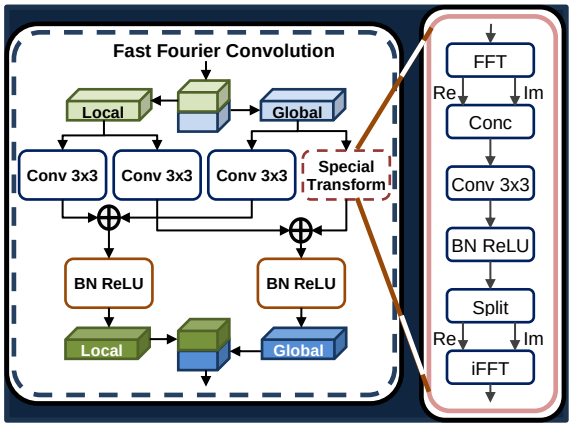


Figure 2 : Fast Fourier Convolution included spectral transform block [9]

The LaMa-Wavelet v1 network applies a single-level 3D Haar wavelet transform [20, 21]. The block of Fourier Unit Structure from the original architecture of the LaMa-Fourier network is replaced by the Wavelet Convolution Block (one level) elaborated by the authors (Figure 3). This block uses a 3D wavelet transform with a Haar wavelet introducing frequency and time analysis different from the Fourier transform. In this case, the decomposition of the obtained coefficients into real and complex parts was also excluded, because the Haar wavelet does not have a complex part. The obtained wavelet coefficients are splitted so that each subband represents a separate feature of the image. Convolutional layer, BN and ReLU activation function are sequentially applied to the results of splitting of coefficients on each level of wavelet transform (Figure 3). The transformed subbands of wavelet coefficients are concatenated and then the inverse discrete wavelet transform (iDWT) is applied.

The LaMa-Wavelet v2 network uses a two-level Haar wavelet transform. The Fourier Unit Structure from the architecture of the LaMa-Fourier network is replaced by the Wavelet Convolution Block (two levels) (Figure 4). This block based on the wavelet CNN architecture [12], where a convolutional layer is applied to the wavelet coefficients at each level of the wavelet transform. Initially, the eight subbands of coefficients at first level of 3D wavelet transform was obtained. Then the first convolutional layer is applied to splitted LLH, LHL, HLL, LHH, HLH, HHL, HHH subbands (Figure 4). The LLL subband is inputted to second level of 3D wavelet transform. The obtained subbands of coefficients are concatted with the first convolutional layer output. The result of concatenation is inputted to second convolutional layer. After that the processed coefficients on the second level of a 3D wavelet transform is concatted with outputs of first convolutional layer. Finally, the inverse 3D wavelet transform on two levels is applied.

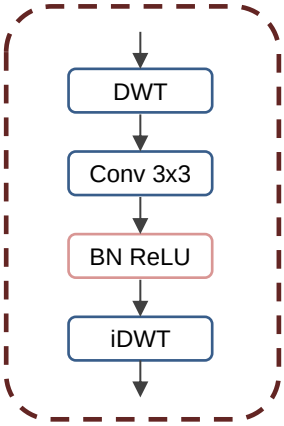


Figure 3: Wavelet Convolution Block

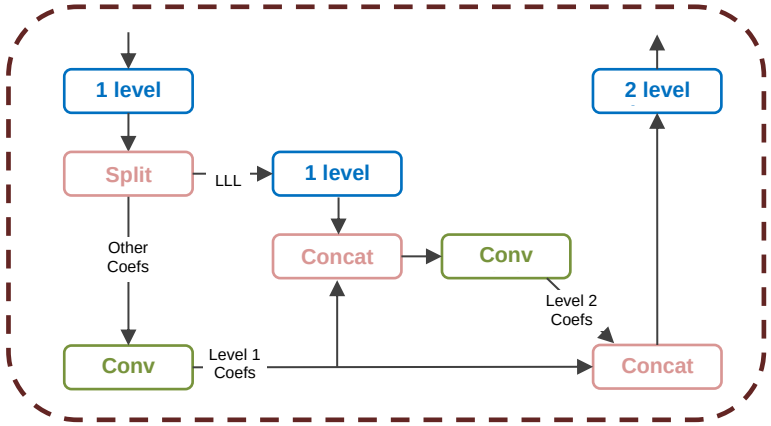


Figure 4: Wavelet CNN based Wavelet Convolution Block

The LaMa-Wavelet v3 network architecture applies a single-level Daubechies wavelet transform. The Fourier Unit Structure block from the original LaMa-Fourier network architecture is replaced by the Wavelet Convolution Block (one level) with Daubechies 4 wavelet (Figure 3). The Daubechies 4 wavelet is chosen because of its ability to capture more complex image features than the Haar wavelet.

In LaMa-Wavelet v4 the Simple Wavelet Convolution Block elaborated by the authors is used instead of Fourier Unit Structure. In this block, 3D wavelet transform of the image on two levels using the Daubechies 4 wavelet was initially performed (Figure 5). The obtained coefficients of 3D wavelet transform are splitted, and convolutional layer, BN and ReLU are sequentially applied to the results of splitting of coefficients on each level of wavelet transform. The obtained subbands of wavelet coefficients are concatenated and the iDWT is applied to them, the result of which is the output of the block.

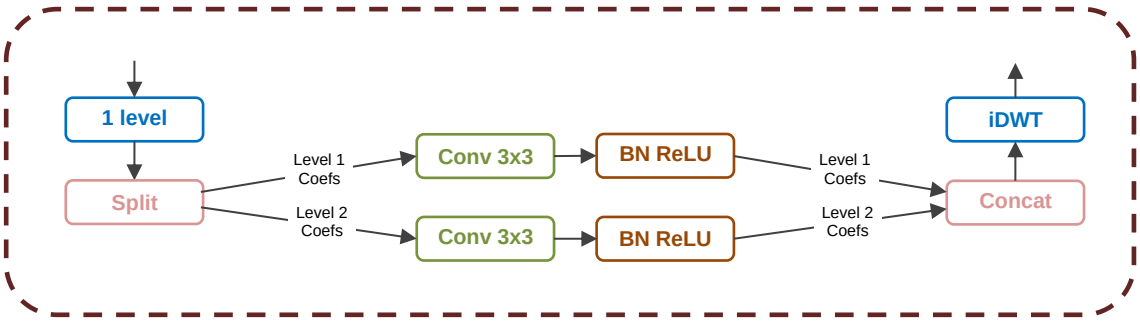


Figure 5: Simple Wavelet Convolution Block

At the third stage of neural architecture search technique the loss function is need to be defined to estimate the training and validation losses. In [7, 9] L_{final} is specially designed to solve the problem of filling large missing regions. In this paper the loss function L_{final} is used which combines pixel loss L_2, perceptual loss L_P and competition loss L_D [7, 9]:

$$L_{final} = kL_2 + aL_P + bL_D \quad (2)$$

where k, a, b are constants.

The mean square error between the original and restored images was used to estimate L_2 pixel loss [7, 9]. The learned perceptual image patch similarity L_P is used to evaluate the perceptual similarity between the restored images and original images using a pre-trained neural network [7, 9].

The discriminator is used to estimate competition loss L_D. This additional CNN is trained in parallel with the basic network to distinguish between real and generated images. Based on this evaluation, the discriminator tunes the basic network coefficients to improve the realism of the generated images. Then, the L_D is the estimation of the error in the global and local textures computed from the discriminator output [7, 9].

5. Experimental Setup

In this section the fourth, fifth and sixth stages of neural architecture search technique are discussed. At the fourth stage of the proposed technique image datasets are selected and pairs (image, mask) is formed to train networks with designed architectures. The 16,000 Places365 and Safebooru database images [22, 23] were scaled to a size of 256x256 pixels and randomly splitted into training and validation sets in the ratio of 95% to 5%. As in [7], for each image it was generated either a mask of 1-4 rectangles with sides of 30-150 pixels, or a mask of 1-5 straight lines 10-200 pixels long, 1-100 pixels wide and with a slope from 0 to 2π. The sizes of the masks were variable, from narrow (10% of the image pixels) to large (80% of the image pixels). This ensures the networks training at different levels of inpainting complexity with masks uniformly coveraging different image areas. To generate a mask of one or another format for specific practical cases, in the described below software the user can draw a mask on the original image.

At the fifth and sixth stages of the proposed technique the LaMa-Fourier and LaMa-Wavelet v1-4 networks were trained and the evaluation of the training results was performed by the Fréchet

inception distance (FID) [24]. FID measures the distance between the feature distributions of real images and images inpainted by the network [24]. The obtained results are compared with existing image inpainting methods based on [25]. Since the FID estimates the overall similarity of the original and inpainted images, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) are applied. These two additional measures evaluates the inpainting of edges and details with the CNN obtained as a result of the neural architecture search [26].

In addition, the impacting of the properties of images from the considered datasets on the result of the neural architecture search was estimated. The measures of homogeneity and uniformity were used to evaluate the saturation of an image with large details and coarse texture [27].

Special software of LaMa network architecture search has been developed for computer experiment. Several key libraries were used to implement LaMa-Wavelet v1-4 in Python, specifically, PyTorch, NumPy, saicinpainting, and others. PyTorch provides tools for designing and training neural networks. It is distinguished by an intuitive API with support for dynamic generation of graphs. This is particularly convenient for the design of the neural network architectures. Also, PyTorch allows using the GPU to accelerate calculation for processing large volumes of data.

For fast mask generation, the mask method of the saicinpainting library was used. To evaluate the results, the pytorch-fid library was also used to obtain the FID score. The psnr and ssim methods of the scikit-image library calculate the PSNR and SSIM, respectively.

The following hardware resources are required to run the program. The operating system is better to use Linux or, alternatively, Windows 10 with support for Windows Subsystem for Linux 2 (WSL2). The processor must support the x86-64 architecture. 1G of disk space is required for the program not taking into account the space for the training set if it is used. The recommended RAM is 16G, the minimum is 8G.

The GPU with CUDA support can be used. Also, Full HD images require 4G graphics memory, alternatively 8G for 2K images, 16G for 4K images. To process data the Google Colab environment was used. It interacted with a pre-configured NVIDIA Tesla T4 GPU, which has 16 GB of GDDR6 memory and 2,560 CUDA cores. Google Drive data storage was used to store the training and testing datasets to easy access to them.

The developed software includes such elaborated Python files as wconv.py and predict.py. The wconv.py module implements spectral blocks of LaMa-Wavelet v1-4 that form the neural architecture search space. DWT calculation was significantly increasing the processing time of LaMa-Wavelet v1-4 networks compared to LaMa-Fourier. Therefore, the ptwt library was used to calculate the DWT coefficients using the GPU.

The wconv.py can process images of given size. So inpainting the larger images after training on small images significantly reduces training time. The predict.py module obtains the image inpainting result using the trained network weights. It requires the following input arguments: the path to the network architecture file; the path to the file with network weights; the path to the folder of source images and their masks; the path to the folder for saving the inpainted images. The user interface of the LaMa network architecture search (Figure 6) is designed based on the T3 template, using Next.js and Tailwind CSS for structuring and styling. The template also includes the trpc framework to ensure reliable communication between the client and the server. React-DaisyUI was used to design the stylized UI components. For image masking the react-canvas-draw library is used, which provides tools for changing the size of the brush, undoing the last stroke and cleaning the image.

The interface displays the loaded image (left) and allows the user to 'paint' a mask directly over it (shown in light green). The Clear button completely erases the drawn mask. The Undo button erases the previous stroke. The Show Mask button turns off or on the display of the drawn mask. The Compare button turns off or on the display of the original image in the result window for quick comparison.

The Brush Size slider changes the size of the mask brush. The Inpaint button starts the image inpainting modules, after which the result is displayed in the window on the right. In addition, the interface includes the ability to select a network for image inpainting, allowing users to search the best option for specific tasks.

After image inpainting the interface shows statistics, including the generation time and the PSNR and SSIM of the inpainted image.

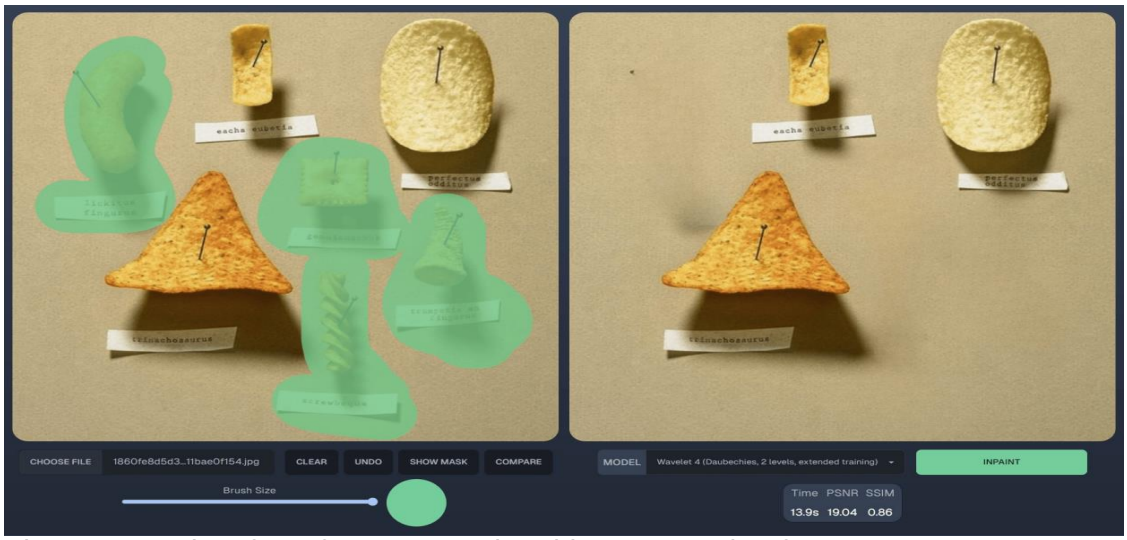


Figure 6: User interface of LaMa network architecture search software

6. Results

In this section the seventh and eighth stages of neural architecture search technique are discussed. The results of the training of the LaMa-Fourier and LaMa-Wavelet v1-4 networks were evaluated using the FID on training and validation sets. The image inpainting time and training epoch time were estimated additionally. Image inpainting time is averaged for a set of 25 images of size 1024x1024 pixels (Table 1).

The dependence of FID from epoch for the LaMa-Wavelet v4 still showed a downward trend after 128 epochs (Figure 7, 8). This indicated the possibility of further reducing of the FID. Therefore the training of the LaMa-Wavelet v4 was continued to 212 epochs. Then the FID of the LaMa-Wavelet was reduced to about 8 on the training set and to 24 on the validation set, approaching the FID of the LaMa-Fourier [7].

Table 1
The neural architecture search results on Places2 dataset [22] 256x256 images

CNN	FID on training set	FID on validation set	Epoch time, minutes	Image inpainting time, seconds
LaMa-Fourier	8.2	25.4	40.1	2.2
LaMa-Wavelet v1	12.1	41.2	94.6	3.8
LaMa-Wavelet v2	45.3	95.0	249.7	13.2
LaMa-Wavelet v3	9.5	37.9	125.3	5.3
LaMa-Wavelet v4	9.2	31.8	148.8	6.6

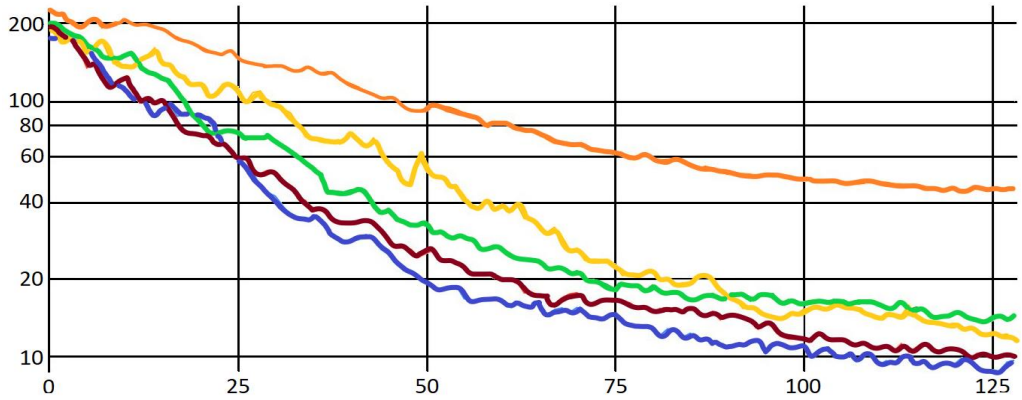


Figure 7: FID score on epochs of LaMa-Fourier network (blue line) and LaMa-Wavelet v1-4 networks (green, orange, yellow and red lines, respectively) on a training set on a logarithmic scale

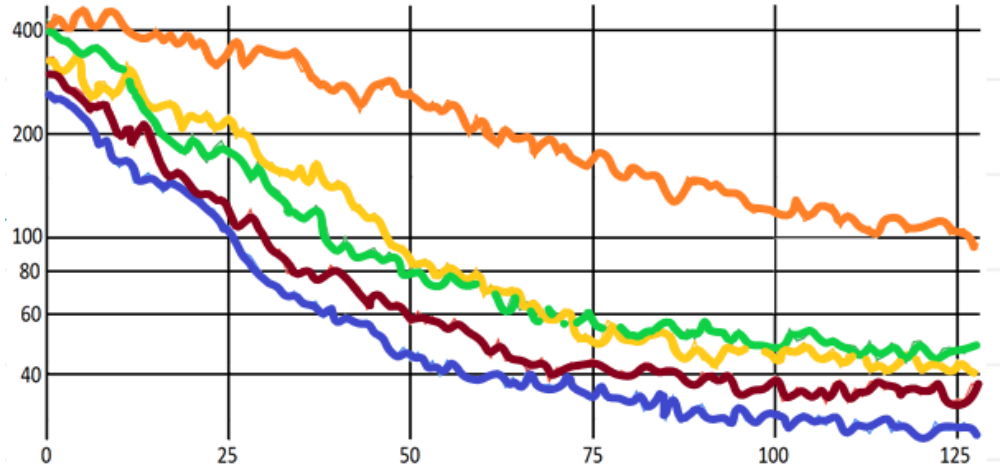
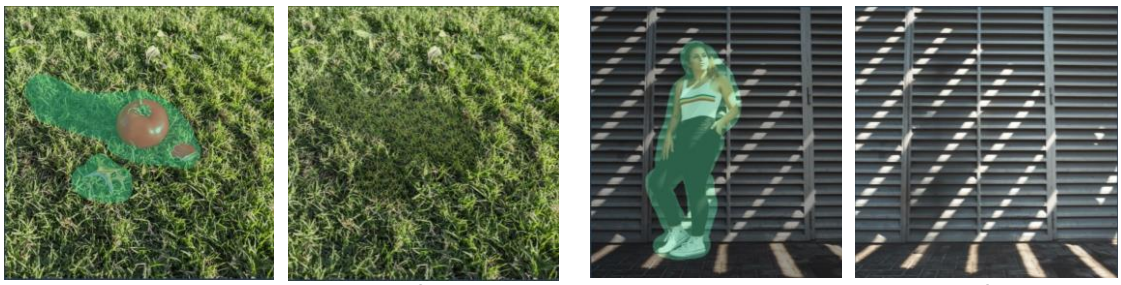


Figure 8: FID score on epochs of LaMa-Fourier network (blue line) and LaMa-Wavelet v1-4 networks (green, orange, yellow and red lines, respectively) on a validation set on a logarithmic scale

When inpanting test set images using the LaMa-Fourier and LaMa-Wavelet v4 networks, it was noticed that the inpainting was better for images containing significant areas of uniform intensity, fine-grained or structural texture (Figure 9, c). Fragments of images, including complex textures or detailed patterns were inpainted worse. For example, the inpainting of grass, leaves, branches, crowd, or thin fabric fibers is difficult for both LaMa-Wavelet v4 and LaMa-Fourier networks (Figure 9, b). Therefore further 50 images with the lowest and highest PSNR, as well as 50 images with the lowest and highest SSIM were selected after inpainting with narrow, medium and large masks. The saturation of the original images with details was estimated with homogeneity and uniformity [27]. These measures were calculated using the gray level co-occurence matrix of the image. In Table 2, 3 homogeneity and uniformity values are given for images inpainted with high and low PSNR and SSIM.



a b c d

Figure 9: Original images with mask (a, c) and images inpainted by LaMa-Wavelet v4 (b, d)

Table 2
The homogeneity of original images which then were inpainted with LaMa-Fourier and LaMa-Wavelet v4 networks

CNN, mask size	Low PSNR	High PSNR	Low SSIM	High SSIM
LaMa-Fourier, narrow masks	0.127	0.595	0.094	0.510
LaMa-Wavelet v4, narrow masks	0.127	0.594	0.101	0.537
LaMa-Fourier, medium masks	0.211	0.357	0.156	0.351
LaMa-Wavelet v4, medium masks	0.157	0.348	0.151	0.351
LaMa-Fourier, large masks	0.234	0.369	0.091	0.343
LaMa-Wavelet v4, large masks	0.261	0.458	0.085	0.387

Table 3
The uniformity of original images which then were inpainted with LaMa-Fourier and LaMa-Wavelet v4 networks

CNN, mask size	Low PSNR	High PSNR	Low SSIM	High SSIM
LaMa-Fourier, narrow masks	2.989	25.126	1.749	12.088
LaMa-Wavelet v4, narrow masks	2.989	25.500	1.735	13.444
LaMa-Fourier, medium masks	28.815	8.669	3.530	8.443
LaMa-Wavelet v4, medium masks	2.182	8.486	3.520	8.443
LaMa-Fourier, large masks	36.558	13.706	0.423	16.563
LaMa-Wavelet v4, large masks	35.284	22.152	0.405	17.606

7. Discussions

Considering the Figures 7, 8 it is noted that the original LaMa-Fourier network is balanced in terms of image inpainting quality and processing time. Training was completed after 128 epochs, providing a reliable baseline for comparison [25]. The validation dataset is used to unbiased estimate the network performance after each epoch. By monitoring the network's performance on the validation set, the training was stopped when the LaMa-Fourier starts to overfit. The training of LaMa-Wavelet v1-4 networks was initially stopped at the same epoch as the training of the LaMa-Fourier network for a consistent comparison. Their learning curves still trend downward, indicating the potential for further performance improvement with continued training. A comparison of training results after epoch 128 (Table 1) showed the following.

The LaMa-Wavelet v2 has shown a worst image inpainting quality and low computational efficiency in comparison with LaMa-Wavelet v1, v3, v4 networks and LaMa-Fourier. Therefore, the interleaving of wavelet transform levels with convolutional layers was decided not to be used, after experiment. This approach negatively affected the computational efficiency of the network, so that it made sense to practically use the LaMa-Wavelet v2. The LaMa-Wavelet v1, v3, v4 networks have reached a level comparable to the original LaMa-Fourier network. These networks have demonstrated a promising balance between image inpainting quality and computational efficiency indicating potential for further optimization. Moreover, the closest result to LaMa-Fourier network was shown by LaMa-Wavelet v4. The LaMa-Wavelet v1, v3 networks showed similar to each other results, which were slightly lower than LaMa-Wavelet v4. Specifically, difference was 26% and 3% for FID on training set, 28% and 19% for FID on validation set respectively (Table 1). The LaMa-Fourier requires less training time and inpaint the image faster then LaMa-Wavelet v1-4. The LaMa-Wavelet v1, v3, v4 are more time consuming. An experiment to estimate the quality of inpainting of edges and fine details of images the LaMa-Wavelet v4 and LaMa-Fourier networks was conducted by the authors in [7]. The PSNR of images inpainted using the LaMa-Wavelet v4 exceeds the results obtained using the LaMa-Fourier network for narrow and medium masks in average by 4.5%, for large masks in average by 6%. The LaMa-Wavelet applying can enhance SSIM by 2–4% depending on a mask size. This issue is covered in more detail in [7]. To analyze the dependence of the quality of generating of details and edges of objects in the image on the properties of the image textures, let's first note the following. To describe image texture properties the texture descriptors of homogeneity and uniformity based on the gray level co-occurence matrix of the image are used. Uniformity increases as the square of the image intensity probabilities, so the less random the image is, the higher its uniformity. Homogeneity characterizes the concentration of the values of the image gray level co-occurence matrix near the main diagonal. A matrix with larger probability values near the diagonal will correspond to a larger value of the homogeneity descriptor. This matrix is typical for an image with a large content of halftones and areas of little changing intensity [27]. So, the results in Tables 2, 3 showed that to fill missing areas of images with large masks, it is preferable to use the LaMa-Fourier network if homogeneity and uniformity is low. If homogeneity and uniformity is high then it better to use the LaMa-Wavelet v4 network to get the inpainted image with high PSNR. To inpaint images with medium masks with high PSNR, it is preferable to use the LaMa-Fourier and LaMa-Wavelet v4 networks if homogeneity is high and uniformity is medium. To inpaint images with narrow masks with high PSNR, it is

preferable to use the LaMa-Fourier and LaMa-Wavelet v4 networks if homogeneity and uniformity is very high. Thus, in the case of large masks, the LaMa-Fourier network is better at inpainting images with more random intensities, while the LaMa-Wavelet v4 network better inpaints images with more halftones and areas of low intensity variation. If the size of the masks is reduced, the ability of both networks to reconstruct images with high detail content increases. However, in the case of narrow masks the both networks is better at inpainting images with areas of low intensity variation.

8. Conclusions

The actual scientific and applied problem of the neural architecture search for the inpainting of the image fine details and object edges has been considered.

The scientific novelty is the technique of neural architecture search for image inpainting proposed. In this way, the new LaMa based network architectures were designed which different by relation between image inpainting time and reconstructed image quality. The image inpainting with large masks based on the LaMa network is improved by applying wavelet transform. Specifically, the quality of filling missing areas with image edges and fine details is increased.

The practical significance of obtained results is that the software realizing the proposed technique of neural architecture search for image inpainting is developed based on LaMa network. Experiments to research image inpainting performance are conducted. The experimental results allow to determine effective conditions for the application of versions of this network in practice. In addition, it was researched the dependence of the quality of generating of details and edges of objects in the image on the properties of the image textures, which can be described by texture descriptors.

Prospects for further research is reducing the computing time by using fast transforms [28, 29] and prediction the effectiveness of the LaMa network depending on the estimated values of image texture descriptors and formulating the recommendations on the LaMa network applications.

References

[1] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, H. Yu, Deep learning for image inpainting: A survey, Pattern Recognition 134.109046 (2023). doi: 10.1016/j.patcog.2022.109046.

[2] D. Kolodochka, M. Polyakova, The research of the quality of filling missing regions of images by methods PatchMatch and LaMa, in: Proceedings of 5th International Scientific and Practical Conference on Modern Research in World Science, SPC "Sci-conf.com.ua", Lviv, Ukraine, 2022, pp. 211–219.

[3] J. Yu, J. Yang, X. Shen, X. Lu, T. S. Huang, Generative image inpainting with contextual attention, in: Proceedings of Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE/CVF, Salt Lake City, UT, USA, 2018, pp. 5505–5514. doi: 10.1109/CVPRW.2018.00577.

[4] J. Yu, Z. Lin, J. Yang et al., Free-form image inpainting with gated convolution, in: Proceedings of IEEE/CVF International Conference on Computer Vision, ICCV, IEEE/CVF, Seoul, Korea (South), 2019, pp. 4471–4480. doi: 10.1109/ICCV.2019.00457.

[5] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, M. Ebrahimi, EdgeConnect: structure guided image inpainting using edge prediction, in: Proceedings of IEEE/CVF Computer Vision Workshop, ICCVW, IEEE/CVF, Seoul, Korea (South), 2019, pp. 2462–2468. doi: 10.1109/ICCVW.2019.00408.

[6] S. Zhao, J. Cui, Y. Sheng et al., Large scale image completion via co-modulated generative adversarial networks, in: Proceedings of International Conference on Learning Representations, ICLR, Vienna, Austria, 2021. doi: 10.48550/arXiv. 2103.10428.

[7] D. O. Kolodochka, M. V. Polyakova, LaMa-Wavelet: image inpainting with high quality of fine details and object edges, Radio Electronics, Computer Science, Control 1 (2024) 208–220. doi: 10.15588/1607-3274-2024-1-19.

[8] L. Cao, T. Yang, Y. Wang, B. Yan, Y. Guo, Generator pyramid for high-resolution image inpainting, Complex & Intelligent Systems 9.7553 (2023). doi: 10.1007/s40747-023-01080-w.

[9] R. Suvorov, E. Logacheva, A. Mashikhin et al., Resolution-robust large mask inpainting with Fourier convolutions, in: Proceedings of IEEE Workshop/Winter Conference on Applications

of Computer Vision, WACV, IEEE, Waikoloa, Hawaii, 2022, pp. 2149–2159. doi: 10.1109/WACV51458.2022.00323.
[10] C. White, M. Safari, R. Sukthanker et al., Neural architecture search: insights from 1000 papers. doi: 10.48550/arXiv.2301.08727.
[11] T. Elsken, J. H. Metzen, F. Hutter, Neural architecture search: a survey, Journal of Machine Learning Research 20 (2019) 1–21.
[12] S. Fujieda, K. Takayama, T. Hachisuka, Wavelet convolutional neural networks, 2018. doi: 10.48550/arXiv.1805.08620.
[13] A. Souza Brito, M. B. Vieira, M. L. Andrade, R. Q. Feitosa, G. A. Giraldi, Combining max-pooling and wavelet pooling strategies for semantic image segmentation, Expert Systems with Applications 183.115403 (2021). doi: 10.1016/j.eswa.2021.115403.
[14] A. Hamad, A new pooling layer based on wavelet transform for convolutional neural network, Journal of Advanced Research in Dynamical and Control Systems 24.4 (2020) 76–85. doi:10.5373/JARDCS/V12I4/20201420.
[15] A. Ferrà, E. Aguilar, P. Radeva, Multiple wavelet pooling for CNNs, in: L. Leal-Taixé, S. Roth, (Eds.), Computer Vision – ECCV 2018 Workshops, volume 11132 of Lecture Notes in Computer Science, Springer, Cham, 2019, pp. 671–675. doi: 10.1007/978-3-030-11018-5_55.
[16] O. Herrera, B. Priego, Wavelets as activation functions in neural networks, Journal of Intelligent & Fuzzy Systems 42.5 (2022) 4345–4355. doi: 10.3233/JIFS-219225.
[17] J. W. Liu, F. L. Zuo, Y. X. Guo et al., Research on improved wavelet convolutional wavelet neural networks, Applied Intelligence 51 (2021) 4106–4126. doi: 10.1007/s10489-020-02015-5.
[18] P. Liu, H. Zhang, W. Lian, W. Zuo, Multi-level wavelet convolutional neural networks, IEEE Access 7 (2019) 74973–74985. doi: 10.1109/ACCESS.2019.2921451.
[19] L. Wang, Y. Sun, Image classification using convolutional neural network with wavelet domain inputs, IET Image Processing 16.8 (2022): 2037–2048. doi: 10.1049/ipr2.12466.
[20] I. Daubechies, Ten Lectures on Wavelets, SIAM Press, Philadelphia, 1992.
[21] J. Bobulski, Multimodal face recognition method with two-dimensional hidden Markov model, Bulletin of the Polish Academy of Sciences, Technical Sciences, 65.1 (2017) 121–128. doi: 10.1515/bpasts-2017-0015.
[22] Places365 Scene Recognition Demo. URL: http://places2.csail.mit.edu/.
[23] Safebooru. URL: https://safebooru.org/index.php?page=post&s=list&tags= no_humans+landscape.
[24] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local nash equilibrium, in: Proceedings of 31st Annual Conference on Neural Information Processing Systems, NIPS, Long Beach, California, USA, 2017, pp. 6629–6640. doi: 10.18034/ajase.v8i1.9.
[25] Supplementary material. URL: https://bit.ly/3zhv2rD/lama_supmat_2021.pdf.
[26] U. Sara, M. Akter, M. S. Uddin, Image quality assessment through FSIM, SSIM, MSE and PSNR — a comparative study, Journal of Computer and Communications 7.3 (2019) 8–18. doi: 10.4236/jcc.2019.73002.
[27] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 4th ed., Pearson, New York, NY, 2017.
[28] A. Cariow, J. Papliński, M. Makowska, VLSI-friendly filtering algorithms for deep neural networks, Applied Science, 13.9004 (2023). doi: 10.3390/app13159004.
[29] A. Cariow, G. Cariowa, Minimal filtering algorithms for convolutional neural networks, in: C. van Gulijk, E. Zaitseva (Eds.), Reliability Engineering and Computational Intelligence. Studies in Computational Intelligence, volume 976, Springer, Cham, 2021, pp. 73–88. doi: 10.1007/978-3-030-74556-1_5.