CASCADED NEURAL NETWORKS FOR IMAGE INPAINTING

Shatokhin A.V. (Russian Federation) Email: Shatokhin575@scientifictext.ru

Shatokhin Arsenii Valerievich – Bachelor of Science,
DEPARTMENT OF INFORMATION ANALYTICS AND POLITICAL TECHNOLOGIES,
BAUMAN MOSCOW STATE TECHNICAL UNIVERSITY, MOSCOW

Abstract: *this paper introduces a new resource-saving solution to the problem of retrieving part of an image. Most image-retrieval functions require the ability to extract structured features to understand the context of an unfamiliar environment and realistic image restoration. The proposed model is based on the conceptual art of convolutional architecture and reproductive opposition models. This method allows you to achieve high quality retrieval of a person's face image, while requiring a minimum amount of resources. The generator design of the generator makes it easier to make better use of the resources used by extracting features from the image tower - the more complex earth features are extracted with less modification, which can significantly reduce the resources used. This paper provides a comparative evaluation of the method used, and the competition, which ensures that by presenting cascade generator design ideas, high quality can be obtained without the use of any back-end processing at minimal source costs. This paper also presents an analysis of the key components of the new approach, showing their importance for the proposed approach.*
Keywords: *deep learning, partial image restoration, cascade neural networks, generative adversarial networks.*

КАСКАДНЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ

Шатохин А.В. (Российская Федерация)

Шатохин Арсений Валерьевич – бакалавр наук,
кафедра информационной аналитики и политических технологий,
Московский государственный технический университет им. Н.Э. Баумана, г. Москва

Аннотация: *эта статья представляет новое ресурсосберегающее решение проблемы восстановления части изображения. Большинство функций поиска изображений требуют способности извлекать структурированные признаки для понимания контекста незнакомой среды и восстановления реалистичного изображения. Предлагаемая модель основана на концептуальном искусстве сверточной архитектуры и моделей репродуктивной оппозиции. Этот метод позволяет добиться высокого качества поиска изображения лица человека, требуя при этом минимального количества ресурсов. Конструкция генератора упрощает более эффективное использование ресурсов, извлекаемых из башни изображений - более сложные объекты земли извлекаются с меньшими изменениями, что может значительно сократить используемые ресурсы. В этой статье дается сравнительная оценка используемого метода и конкурентов, что гарантирует, что путем представления идей проектирования каскадных генераторов можно получить высокое качество без использования какой-либо внутренней обработки при минимальных затратах на источник. В этом документе также представлен анализ ключевых компонентов нового подхода, показывающий их важность для предлагаемого подхода.*
Ключевые слова: *глубокое обучение, частичное восстановление изображений, каскадные нейронные сети, генеративные состязательные сети.*

1. Introduction

Convolutional neural networks have managed to solve many problems in the field of computer vision. One of these tasks, where convolutional networks have become actively used recently, is the task of restoring a part of an image (image inpainting). The problem of restoring a part of an image has many applications, such as removing unwanted objects from an image, changing the visual attributes of scene objects, and restoring images. For example, using these methods you can remove watermarks from photos, unwanted text, an extra person in the background, or remove scratches from old photos. An example of an applied problem of removing an object from a scene is shown in Figure 1.

Fig. 1. An example of using the task of restoring a part of an image - removing unwanted objects from an image. In the left photo, outlined the area is considered a recovery area. The result is shown on the right

For the case of restoration (deletion) of a large complex-structured part of an object, the main difficulty is the extraction of features to determine the context of the restored area. Moreover, in such a problem there is no exact estimate of the quality of the result of the method - if the unknown region is very large, then it can be reconstructed in many ways. The only criterion for success in such a task is the plausibility of the obtained recovery.

Most of the recent best practices for working with complexly structured objects are based on ideas from generative adversarial models (GAN) [1]. They solve both above problems: a generator is a large convolutional neural network that can extract highly complex features. To assess the likelihood of the result, a second convolutional neural network is used - a discriminator with a tunable (trainable) validation metric of areas reconstructed by the generator. The main problems of such approaches are most often: the complexity of training (such models can take months to learn), a long time of work and the amount of resources consumed during training and application. All these problems make such methods very impractical for real-world applications. For example, it is very difficult to use such solutions on mobile devices, which are still very limited in resources, especially for the case of high resolution.

The paper proposes a new solution for the problem of recovering a part of an image in high resolution. The new method consumes a small amount of resources and allows achieving high visual quality for solving the problem. The method is based on the approach of cascading neural networks, first proposed for solving the problem of semantic segmentation [2]. The main idea of the method is to build an image pyramid and independently extract features in different resolutions. At the same time, most of the complex semantic features are extracted at the lower level, which makes it possible to efficiently use the consumed resources, and for the upper levels of the pyramid, only local features are extracted, which are necessary to refine the solution obtained in a low resolution.

The work shows that it is possible to achieve a high quality solution to the complex problem of restoring human faces, where, without a doubt, faces are objects with a complex structure and great variability, even with very strict restrictions on consumed resources: the number of parameters of the resulting model is much less than that of all advanced methods solving this problem, for training the method does not require many video cards and months of time. At the same time, the quality of the method is not inferior to competing methods in this area. The proposed method was tested on CelebA-HQ data [3].

Section 2 provides an overview of best practices for solving the problem of restoring part of an image and identifies the problems of each of the approaches. Chapter 3 details the developed method based on the idea of cascading convolutional networks. Section 4.2 provides an experimental comparison of the proposed method with one of the competing approaches, and Section 4.3 presents an analysis of the main components of the method described in the work.

Overview of relevant methods

There are two large families of methods for solving the problem of image restoration. The first are traditional (non-learning) computer vision methods based on five searches for similar parts of the image (exemplar-based) or contraction of boundaries (diffusion-based). The second group is trainable neural network models.

Traditional methods for solving the problem

One of the most famous traditional approaches to solving the problem is the “PatchMatch” method, as well as its various modifications [4, 5, 6]. For example, the method [5] uses the idea of pyramidal image reconstruction. Starting from the lowest resolution, the algorithm reconstructs the unknown area, gradually increasing the image resolution. Having obtained an approximation at a low resolution, the method refines the solution at a higher resolution. In the approach proposed by the authors of the article [6], the authors propose to statistically distinguish

several dominant shifts of the known regions, combining which can effectively restore the unknown region.

The problem with all such approaches is the inability to recover complex-structured objects. Moreover, the methods are capable of reconstructing the unknown part of the image only by combining the known regions and are not able to generate details that do not exist in the image, which may be necessary for realistic reconstruction of the unknown part.

Neural network methods for solving the problem

One of the first solutions using the idea of generative models for the problem of reconstructing a part of an image was proposed in [7]. Although the method potentially allows one to solve one of the most important problems of non-learning models the restoration of a part of an object by generating rather than mixing parts close in context, it is much inferior in quality to many newer methods, especially for complexly structured objects, for example, for human faces.

In [8], the authors proposed to use the attention layer in the space of features extracted by a convolutional neural network to better search for similar areas in the image by context to the reconstructed one. The model consists of two parts - for a rough approximation and for a refinement. In this regard, the resulting model is very large and requires a lot of resources even just for launching, not to mention training. At the same time, the authors also make the assumption that the deleted part can be restored with known parts of the image, which is far from always true in the case of the task of restoring a part of the face.

In the closest work to our GMCNN studies [9], the authors proposed using a generator architecture with several independent parts to extract features at different resolutions with different core sizes. The main problem with this approach is that all parts of the model built for feature extraction (encoders) have almost the same architecture (differ only in core sizes) and extract features independently of each other. After extraction, all features are combined and processed together, which does not allow building resource-optimal architectures and smoothly moving from global features to local ones.

In the model proposed in our work, the part responsible for extracting features in low resolution is much deeper than in high resolution. After extraction, the features are combined sequentially from the lowest resolution to the highest. For more efficient resource consumption, we used the idea of split convolutional blocks [11]. All this led to a new approach, which is not inferior, and even surpasses the model proposed in [9] in terms of visual quality, while significantly gaining in terms of consumed resources.

There are also many interesting modifications to the task of restoring a part of an image. For example, in [8], the authors proposed a method for restoring a part of an object in an image with conditioning on additional attributes: gender, presence or absence of a smile. The high-resolution visual results of the method presented in the work are very impressive, but one of the most important disadvantages of the method is the cumbersomeness and time of training the model - more than 3 weeks of training on the Titan Xp GPU. The reason was that the authors used the Progressive Growing approach [10]. Such models are too large to be used when resources are scarce. Our research can contribute to the further development and improvement of such approaches.

Conclusions and results by chapter

This chapter gave an overview of existing methods for solving the problem of restoring a part of an image. The drawbacks of each of the two families of approaches are pointed out: non-learning methods are not able to restore

complex-structured parts of images, and neural network approaches showing high visual quality require too many resources.

The following chapters describe the developed method and provide an experimental comparison of the new approach with the GMCNN method [9], which is a direct competitor. of the presented approach.

Proposed method

The proposed method is based on the idea of competitive neural networks: to construct a solution, a pair of simultaneously trained models is used - a generator and a discriminator. A model with a cascade architecture was used as a generator. The generator accepts a three-channel image as input X, as well as a single channel mask M (with a value equal to 1 for the unknown area, 0 for the known). As the output of the generator returns the restored image X rec, the plausibility of which is assessed by the second model - the discriminator. Subsequent parts of this section provide detailed description of the cascade architecture of the generator with all its components, the used model of the discriminator is also described.

Cascaded generator architecture

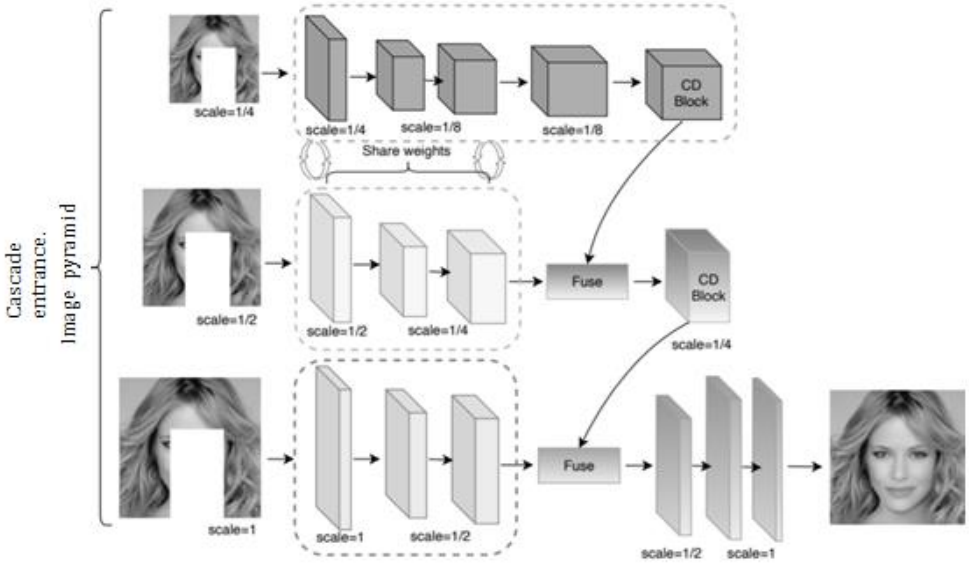

Fig. 2. Diagram of a cascade generator. "CD Block" - a cascading block of increasing the scope, "Fuse" - a merge block, "scale" - a scale width and height of features

The cascade architecture of the generator used to solve the problem of restoring a part of the face is shown in Figure 2. The model accepts a pyramid with three times scale of the original image for restoration. First, the features are independently extracted for each level of the pyramid. For the highest resolution, the number of extracted features is the smallest; such features are the most local, while for the lowest resolution the extracted features are more complex and global. This approach with cascading feature extraction allows efficient use of resources (in high resolution the least number of operations), while due to the cascading model architecture, features are explicitly separated into local and global. There is only one downsampling at each level of the model, so the “CD” box is used to increase the receptive field the model. The block is described in more detail in the next chapter.

After features are extracted, they are gradually merged and refined, starting from low resolution, and ending with high resolution. The fusion block "Fuse" is responsible for the merge operation, the block diagram is shown in Figure 2. After the merge, several convolutional blocks follow at each level. In low and medium resolution, after merging, there is a “CD” block, for high resolution there is no such block since it consumes too many resources.

All feature extraction blocks consist of "Residual blocks" [12], where separable convolutions are used instead of usual convolutions for faster learning, less resource consumption, and less overfitting. Changing the resolution of features occurs by applying a downscaling (or upscaling) operation using the nearest-neighbor scaling.

After getting the predictions of the model G (X, M) final decision X rec obtained by mixing G (X, M) and X in the following way:

$$X^{rec} = G(X, M) \odot M + X \odot (1 - M).$$

Where*- the operation of pixel-by-pixel multiplication.

The work did not use any post-processing to refine the solution. The results of the work are shown in section 4.2.1.

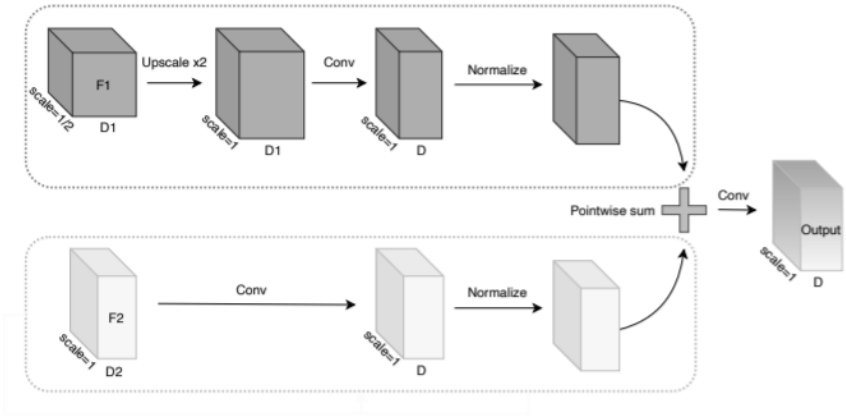

Figure: 3: Diagram of a merge block. F1 - Extracted tags at a lower resolution than F2. D1 and D2 hinterland of feature maps F1 and F2. "Output" is the result of the merge. D is the hinterland of the feature map for the merge result. "Conv" is a convolutional block. "Normalize" - normalization block. "Pointwise sum" is a block of per-pixel addition

Cascading FOV Magnification Unit

To solve the problem of restoring a part of the face, the model must be able to extract global features to understand the context. For example, you need to know which part of the face was removed or whether the person was wearing glasses.

In the proposed architecture (Figure 3), there are only a few downscaling blocks. The small number of downsamples is caused by the fact that it becomes more difficult for the model to recover fine details in the image after a large number of downsamples, especially with not too deep feature maps. With a limitation on consumed resources, deep feature maps simply cannot be built.

To increase the field of view (to obtain more global features), a cascading block of increasing the field of view ("CD block") was used. The block diagram is shown in Figure 4. The block is based on the idea of “dilated convolutions" [13]. The first stage of the block is to extract features by applying convolutional blocks with different values of the "dilation" extension parameter: 1, 2, 4 and 8. For the smallest value of the parameter, the features are the most local and are needed to clarify more global features with the largest value of the expansion parameter, since at the maximum value of the parameter, although the features are more global, they have a sparse and imprecise structure. The depth of feature maps for each parameter value is exactly 4 times less than the initial one (the total depth of all four cards drawn matches the input). After that, the extracted cards are gradually merged, starting from the lower value of the parameter, ending with the maximum. After all merges, the extracted features supplement the original ones (complicate) by pixel-by-pixel addition.

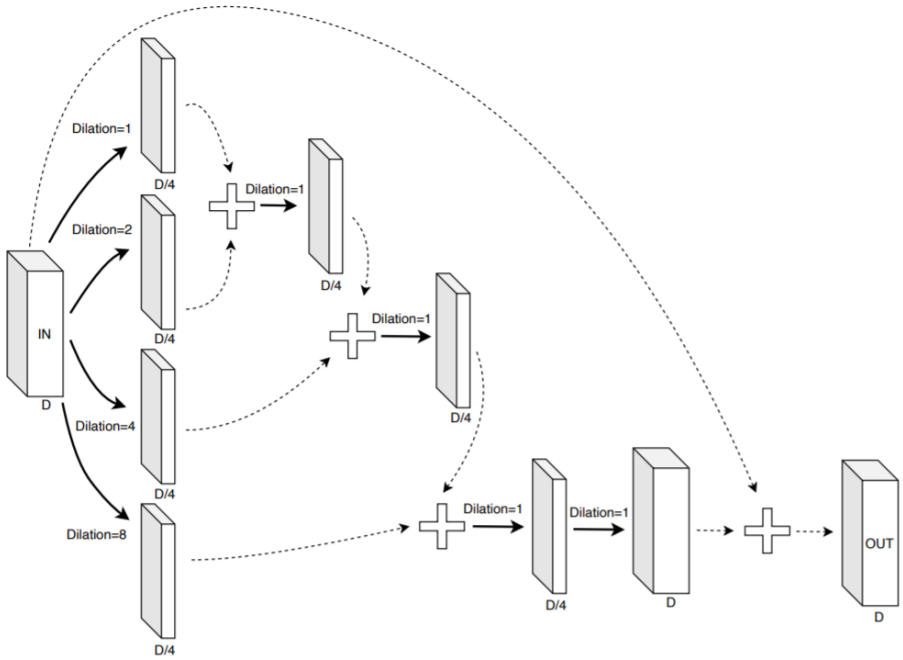

Fig. 4. Diagram of the field of view magnification block "IN" - block input, "OUT" -the result of applying the block. Bold arrows indicate convolution operations with the specified extension parameter "Dilation". The dotted arrows show where the output of the previous operation is transferred. Plus, the operation of per-pixel addition "pointwise sum" is marked

Loss function for generator

Competitive part

The main component of the loss function is the adversarial component

$$\mathcal{L}_{adv}(\theta) = \mathop{\mathbb{E}}_{X \in \mathcal{P}_{corrupted}} \{\log(1 - D_\tau(G_\theta(X)))\}$$

Where θ - trainable parameters of the generator, P corrupted - distribution of examples for recovery, D τ - discriminator model with parameters τ.

Feature Map Mismatch Error

The next component of the loss function is the error of mismatch of feature maps (perceptual loss) [14]. It helps the generator to start learning when the discriminator is untrained. The error is calculated only for the area to be reconstructed and the known area does not participate in the value of the loss function (the size of the mask should not affect the absolute value of the function losses).

$$\mathcal{L}_{feat} = \sum_{j=1}^{j=5} \left\| M \odot \left(\phi_j(X_\theta^{rec}) - \phi_j(X^{real}) \right) \right\|_2^2$$

Where φ j (X) - activation (signs) of the layer RELU j_ 2 for image X the pretrained VGG19 model [15] on the ImageNet problem [16], M - mask of the restored area for example X.

Pixel Mismatch Error

For small areas, a pixel-by-pixel metric can also be useful, so a mismatch error was used (reconstruction loss).

$$\mathcal{L}_{rec}(\theta) = \left\| M \odot \left(X_\theta^{rec} - X^{real} \right) \right\|_1$$

Final loss function

The resulting loss function for the generator is as follows

$$\mathcal{L}_{total}(\theta) = \lambda_{adv}\mathcal{L}_{adv}(\theta) + \lambda_{feat}\mathcal{L}_{feat}(\theta) + \lambda_{rec}\mathcal{L}_{rec}(\theta)$$

Where λ adv, λ feat, λ rec - hyperparameters that regulate the weight of each of the components.

Discriminator model. Cascading discriminator

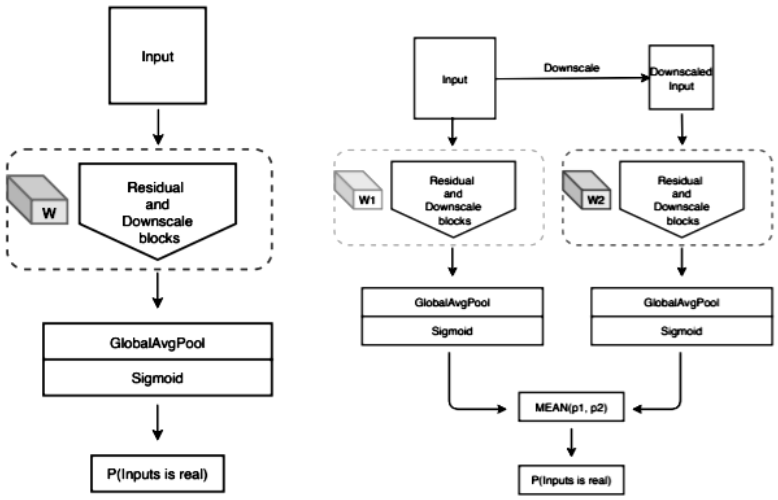

Fig. 5. Figure "a" shows the basic discriminator circuit, figure "b" cascade discriminator. The weights for different branches of the cascading discriminator are different

Basic and cascading architecture

The discriminator is a convolutional network that accepts an image as input, and outputs some estimate of the realism for this image. In our case, the discriminator predicts the probability that the input image is real. Very often the discriminator is stronger than the generator. The discriminator manages to learn too quickly at the beginning of training to distinguish real examples from generated ones, which does not allow the generator to learn. To combat this problem, different regularizations are used for the discriminator. In this work, the idea of spectral normalization was used to regularize the discriminator [17]. As well as the generator, the discriminator model consists of the residual convolutional blocks "Residual blocks". In contrast to the generator, the discriminator model does not have a scope increase unit and there is no resolution increase unit.

At the initial stage of training, the basic discriminator model 3.4a was used. At the stage of additional training of the generator (more about additional training in Section 3.5), the idea of a cascade discriminator proposed in [18] was used. The concept of a cascade discriminator is that an image is supplied to it as an input, a pyramid of images is built from the image, by decreasing the dimension for each next level. Then each resulting image is independently processed by a separate discriminator branch (each branch has its own weights). After that, the final prediction is obtained by averaging the results of all branches. This approach allows us to explicitly divide the discriminator features into more local and more global ones: for a low-resolution image, the field of view is larger than for a high-resolution image, so the features are more global. In the case of the high-resolution branch, features aim to validate finer image details. We used a pyramid for the discriminator of two levels - a branch for the original image and for a halved one.

The loss function for the discriminator

The loss function for the discriminator consists only of a competing component

$$\mathcal{L}_{adv}(\tau) = -\frac{1}{2}\Big[\mathop{\mathbb{E}}_{X\in\mathcal{P}_{real}}\{\log(D_\tau(G_\theta(X)))\} + \mathop{\mathbb{E}}_{X\in\mathcal{P}_{corrupted}}\{\log(1 - D_\tau(G_\theta(X)))\}\Big]$$

Where τ - discriminator parameters, D τ - discriminator model, P real - real Images, P corrupted - recovered images by generator G θ.

Training details

As described in section 3.4, the generator model was trained in two stages. At the first stage, the basic

discriminator was used to obtain an approximate solution. In order to achieve better visual results, a retraining stage with a cascade discriminator was used. The influence of additional training on the quality of the resulting model is investigated in Section 4.3.

For both stages of training, the "Adam" optimizer was used [19]. Before the start of the retraining stage, the optimizer parameters were reset. Also, the weights of the discriminator from the first stage did not participate in the second in any way; all the weights of the discriminator at the second stage were trained anew.

The learning epochs of the generator and discriminator alternated. One training epoch of the generator consisted of 10 batches, and the discriminator of 5. The location of the mask was randomly chosen evenly throughout the picture. For all examples, the mask was always rectangular. Mask size was sampled from a uniform distribution U [low; 1] regardless of each side, where the parameter value low = 0.2 for the initial stage and low = 0.5 for the stage of additional training.

Conclusions and results by chapter

The chapter presented a new method for solving the problem of restoring a part of an image. All components of the approach are described: a cascade generator model, a cascade block to increase the visibility area, a discriminator model, loss functions for the generator and discriminator. The motivation for using each of these components is given, as well as the details of training the models.

The developed approach shows very high results for the task of restoring a part of the face, while consuming a small amount of resources. The next chapter presents an experimental comparison of the obtained method with the GMCNN method [9], as well as an analysis of the components of the developed approach.

Results of the method. Experiments

Description of test data

For the experiments, CelebA-HQ data [3] with celebrity faces were used. All metrics and comparisons with the GMCNN method are presented for test data at 256x256 resolution. Metrics are calculated on the validation part of the dataset (deferred sample of 100 examples). For each example, the mask is a rectangle with a random top-left corner position. The size of the sides of the rectangle is also random and occupies at least 35 percent of the original width and height of the image. No post-processing was used in the work, the results of image restoration coincide with the model output.

Comparison of the method with a competing approach

For comparison, the GMCNN method [9] was chosen, which is the closest to the set task - it uses a generative approach and does not consume too many resources. To obtain the results of the GMCNN method, the pretrained by the authors was used model. No post-processing of the results for GMCNN was used as in the original work.

Table 1. Metrics for GMCNN and the proposed method. For PSNR and SSIM values of the metrics are given, for AB testing the average percentage of preferences is the percentage of selected pairs in which the method result was preferable for the assessor

	GMCNN	Proposed method
PSNR	23.4393	23.8215
SSIM	0.8866	0.8872
AB testing	19.5	80.5%

Examples of how methods work

Figure 6 shows several examples of the work of the proposed method and the GMCNN method. From the examples given, it can be seen that the approach proposed in the work is more stable than GMCNN, it has much less artifacts on the background and on the face than GMCNN, and the boundaries of the restored area are less noticeable. GMCNN suffers from all the above problems, but in some examples GMCNN restores some parts of the face in more detail - the eyes of the girl from the last example in the figure 6 are more detailed.

Comparison of numerical metrics

The standard metrics for this task were used as metrics - PSNR (peak signal to noise ratio), SSIM (structural similarity). The values of the metrics for the two considered methods, calculated on the validation set, are shown in Table 1. The higher the metric value, the higher the quality.

In the problem of restoring a part of a face, the result may look realistic, but the output of the method may differ significantly from the original image if the restored area is large. Since the PSNR and SSIM metrics operate on pixel-by-pixel similarities between the true image and the reconstructed image, they may be completely unrepresentative for cases of large, reconstructed areas. In this regard, just as in [9], one more metric “AB testing” was used.

The idea of AB testing was that the assessor was shown two photos - the result of GMCNN and the result of the method proposed in the work. After that, he chose more.

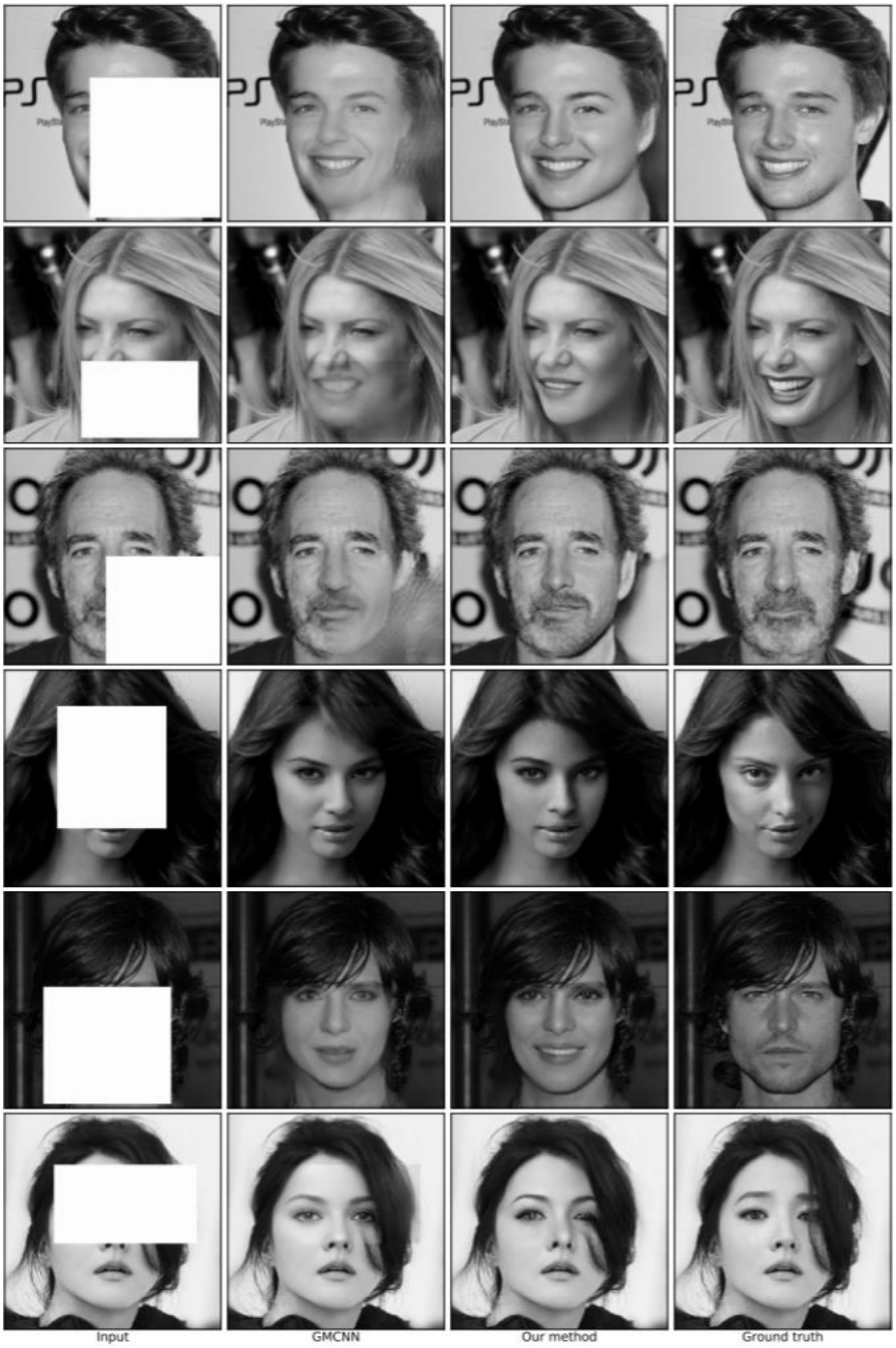

Fig. 6. Examples of how facial reconstruction works for the GMCNN method and the proposed method. "Input" - the image to be restored, "GMCNN" - the results of the GMCNN method, "Our method" - the results of the proposed method, "Ground truth" - the original picture. All images are from the validation set and have resolution 256 × 256

A photograph that is realistic in his opinion. For testing, the same pairs were used for which the PSNR and SSIM metrics were calculated (100 photos from the validation set). All 100 pairs were shown to each assessor. The assessors passed the test independently and with no time limit to choose from. For each assessor, a random order of displaying pairs of photographs was chosen; moreover, the result of each of the methods could appear both on the left and on the right side. Ten people aged 20 to 30 years were selected as assessors. Most of the selected users work in the field of computer vision and are familiar with the problem being solved, several users have never heard of the problem being solved. The final metric for the AB test is the average value over all the user of the percentage of preference of the method result over the opponent's result.

The AB test results are shown in Table 1. On average, 4 out of 5 couples chose the result of the proposed method. Such a high percentage confirms that the method proposed in this work is more stable than GMCNN - most of the images reconstructed by our method are practically free of artifacts that appear because of GMCNN operation. The less obvious boundaries of the restored area for the results of our method also became an important factor in the choice. Among all assessors, the lowest percentage of preferences for the developed method is 66, the highest is 87.

Resources

One of the most important advantages of our solution is the low amount of consumed resources. The resulting model is very lightweight and extremely fast.

Table 2 shows the data on the consumed resources for the two compared methods. The models were tested on a GeForce GTX 1080 GPU as well as an Intel Xeon (R) CPU E5-2620 v3 2.40GHz CPU.

The number of parameters of the GMCNN model is more than 36 times greater than the number of parameters of the model proposed in the work. Such a large number of parameters leads to problems with using the model, for example, in mobile applications. The model presented in the work is more lightweight (the model in fl oat32 weighs only 1.3 MB). The GMCNN method is slightly ahead of the method proposed in the work in terms of operating time on a server GPU (insignificantly since both methods work very quickly), while noticeably inferior in operating time on a CPU. This fact can be explained by the fact that the GMCNN model is not adapted to work on more mobile devices - on devices that lack powerful graphics processors. So, for example, the model contains many inseparable convolutions with large kernels (5 or 7), which makes it not well suited for low/mid-performance CPUs

Table 2. Resources consumed for the compared methods: number of parameters, running time on a high-performance GPU, and CPU for 256x256 images

	GMCNN	**Proposed method**
Number of parameters	12.562 M	0.344 M
Mean GPU time	0.013s	0.021s
Mean CPU time	3.359s	1.678s

Analysis of the main components of the method

This section shows the contribution of several main components of the method, gradually adding which it was possible to achieve the results presented in the work. These components are: adding a cascading block for increasing the field of view (CD block), which was presented in Section 3.2, refusal to normalize in a merge block, as well as the use of the stage of additional training with a cascade discriminator (Section 3.4). All of the above modifications were added sequentially. Figure 10 shows examples of how the method works after adding each of the components described in this section. In the following chapters, for each of the considered modifications, the motivation for its use is given and several examples demonstrate the change in the quality of recovery after adding only this modification.

Influence of cascading field of view enlargement unit

As mentioned earlier in the work, to solve the problem of restoring a part of a model's face, it is necessary to be able to extract complex global features (features from large areas of the image). This requires the model itself to have a large receptive field. At the very beginning of the work, the cascade block for increasing the visibility was not used, the results were very blurry, the model could not draw the eye or glasses symmetrically. The use of multiple cascading blocks to increase the field of view at low and medium resolutions has helped make a significant contribution to solving these problems. Figure 7 shows examples of the operation of one of the very first trained models without a cascading block of increasing the visibility area, as well as with the addition of several such blocks to the model.

Abandoning normalization in a merge block

The next step after adding the CD block was to examine the merge block.

Fig. 7. (a) - the original image for restoration, (b) - the result of the base model without a cascade zoom unit, (c) - the result after adding a cascade zoom unit to the base model field of view

In the original work with the cascade model for semantic segmentation [2], the fusion block used batch normalization of each of the two feature maps before merging them. At the beginning of this work, the method also used batch normalization. The decision was made to abandon normalization in the merge block. The main motive was the fact that the batch normalization block, mixing information about the features of different examples (trying to make their statistics more similar), can lose the individual features of each of the examples. Because of this, the boundary between the restored and the known area may be more pronounced (the colors may differ more). Failure to normalize resulted in smoother colors, less visible boundaries of the restored area.

A study was also carried out to replace batch normalization with normalization according to one example (instance norm) [20]. The model trained with normalization from one example is less significant, but inferior in visual quality to the model without normalization.

Additional training using the cascade discriminator

Section 3.4 describes the idea of retraining a model with a cascading discriminator architecture. This additional training allowed us to achieve better visual results for the restoration of large areas of the face, as well as better detail. Figure 9 shows examples of the method operation without additional training and with additional training using a cascade criminals, demonstrating the benefits of such additional training.

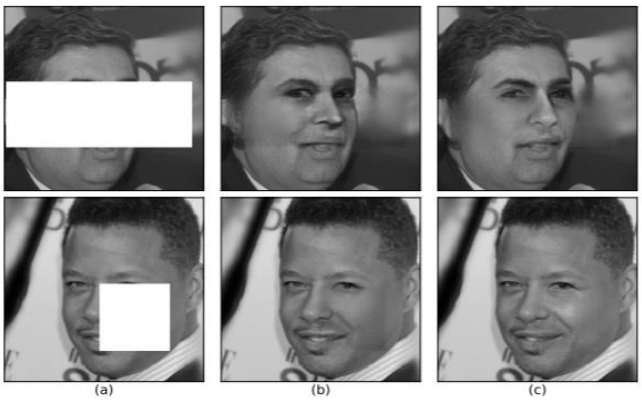

Fig. 8. (a) - original image for restoration, (b) - result of work models c using batch normalization in a merge block, (c) – result work of the model without using batch normalization in the merge block

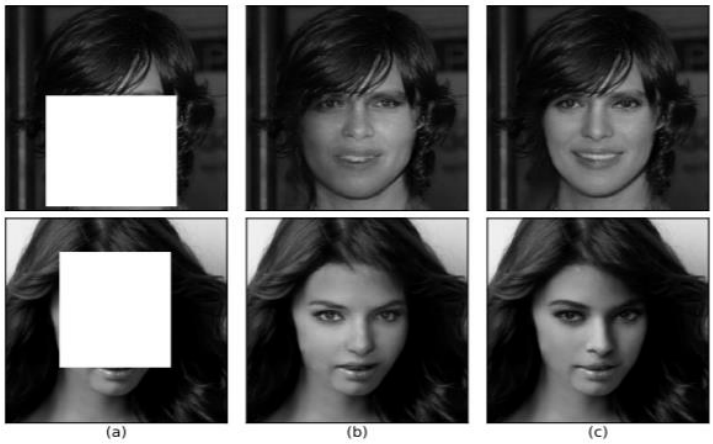

Fig. 9. (a) - the initial image for restoration, (b) - the result of the model without additional training, (c) - the result of the model with additional training with a cascade discriminator

Results for very high resolution

We also trained the model for a very high resolution of 1024x1024. The architecture and number of parameters of the generator model remained the same as for the 256x256 resolution. The retraining stage with a cascade discriminator was also used.

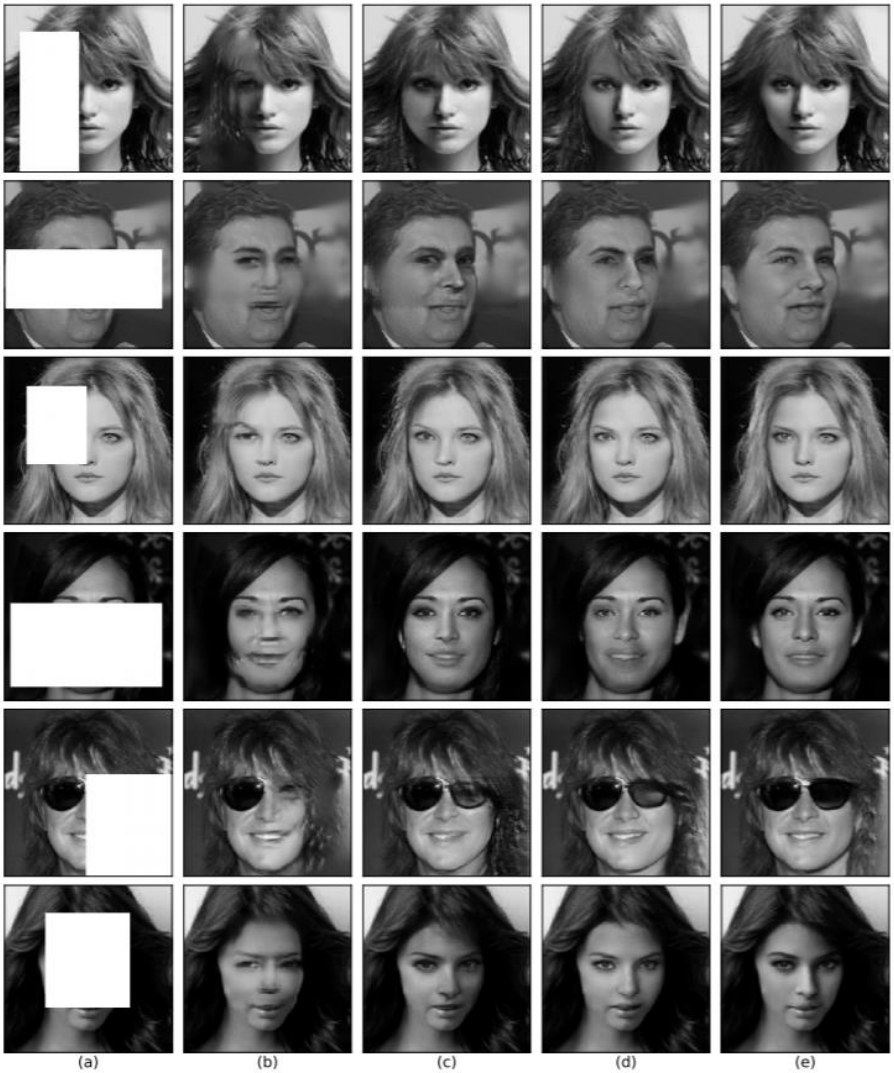

Fig. 10. Stages of model improvement: (a) - images for recovery, (b) - results of the basic waterfall model, (c) results after adding a CD block, (d) - results after abandoning batch normalization, (e) - results after additional training with a cascade discriminator (final version)

We were interested to know how such a lightweight model can cope with very high-resolution faces, whether it has enough field of view to adequately restore the image. Several examples of how the trained model works are shown in Figure 11. Even such a lightweight model (the number of model parameters is less than even the number of image pixels) copes very well with restoring a human face. It can be concluded that the recovered by the generator features are very representative for 1024 resolution as well. In contrast to the 256x256 resolution, more small noticeable artifacts appeared on the model results (they are visible with a strong approach). The nature of these artifacts has not yet been investigated. We assume that this problem can be solved by adding a loss function responsible for smoothness and increasing the number of pyramid levels in the cascade architecture of the generator and discriminator.

Fig. 11. Several examples of how the developed method works for an ultra-high resolution of 1024x1024 on a validation set. Number of parameters the model is less than the number of pixels in the image

Conclusions and results by chapter

In this chapter, experiments were presented to assess the quality of the developed method. Experiments have shown that our method is superior in quality to the competing GMCNN method. At the same time, the developed model is lighter than the GMCNN model and works faster in the absence of high-performance GPUs.

The chapter provides an analysis of the main components of the developed method in order to demonstrate their impact on the final quality of the model. All three investigated components: a block for increasing the scope, refusal to normalize in a merge block, and additional training with a cascade discriminator significantly improved the quality of the final solution. The chapter also shows the model results for a very high-resolution face image of 1024x1024. The above results prove that even the current small model is capable of showing very high visual results without any improvements, while the number of model parameters is even less than the number of pixels in the input image.

Conclusion

The paper deals with the problem of restoring a part of an image. To solve the problem of restoring a part of a face, a method has been developed based on the idea of competitive generative models, as well as a cascade architecture for a convolutional network. The studies carried out show that the cascade architecture of the generator makes it possible to achieve a high quality of the problem being solved, while the obtained method requires a very small amount of consumed resources.

Also, an important contribution of this work was the analysis of the influence of several of the most important components of the proposed method on the quality of the developed approach: the effect of the cascade block of increasing the visibility area, the use of additional training of the model with a cascade discriminator.

Further research will be analyzing the behavior of the method for irregular shapes of the restored regions (not rectangular regions), improving detail and reducing small artifacts for high resolution, as well as applying the method to other problems of restoring a part of the image.

References / Список литературы

1. *Goodfellow Ian J., Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron C. and Bengio Yoshua*. Generative adversarial nets. NIPS, 2014.
2. *Zhao H., Qi X., Shen X., Shi J.* Icnet for real-time semantic segmentation on high-resolution images, arXiv:1704.08545, 2017.
3. *Tero Karras, Timo Aila, Samuli Laine and Jaakko Lehtinen*, 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196, 2017.
4. *Barnes C., Shechtman E., Finkelstein A. and Goldman D.* Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, 2009.
5. *Fedorov Vadim, Facciolo Gabriele and Arias Pablo*. “Variational framework for non-local inpainting”, Image

Processing Online. Vol. 5. Pp. 362–386, 2015.
6. *He K. and Sun J.* Statistics of patch offsets for image completion. In ECCV. Pages 16–29. Springer, 2012.
7. *Pathak D., Krahenbuhl P., Donahue J., Darrell T. and Efros A.A.* Context encoders: Feature learning by inpainting. In CVPR, pages 2536–2544, 2016.
8. *Yu J., Lin Z., Yang J., Shen X., Lu X. and Huang T.S.* Generative image inpainting with contextual attention. arXiv preprint arXiv:1801.07892, 2018.
9. *Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia.* Image inpainting via generative multi-column convolutional neural networks. In NeurIPS, 2018.
10. *Karras Tero, Aila Timo, Laine Samuli and Lehtinen Jaakko*, 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.
11. *Howard Andrew G., Menglong Zhu, Bo Chen, Kalenichenko Dmitry, Weijun Wang, Weyand Tobias, Andreetto Marco and Hartwig Adam.* Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017. 2, 4, 5, 6.
12. *He K., Zhang X., Ren S. and Sun J.* Deep residual learning for image recognition. In CVPR, 2016. 1, 2, 3, 4, 5, 6.
13. *Yu F. and Koltun V.* Multi-scale context aggregation by dilated convolutions. In ICLR, 2016.
14. *Johnson J., Alahi A. and Fei-Fei L.* Perceptual losses for real-time style transfer and super- resolution. 2016
15. *Simonyan Karen and Zisserman Andrew*, 2014. Very deep convolutional networks for large- scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
16. *Krizhevsky Alex, Sutskever Ilya and Hinton Geoffrey E.*, 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Neural Information Processing Systems (NIPS). 1106–1114.
17. *Takeru Miyato, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida*. Spectral normalization for generative adversarial networks. In ICLR, 2018.
18. *Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Tao Andrew, Kautz Jan and Catanzaro Bryan.* High-resolution image synthesis and semantic manipulation with conditional GANs. CoRR, abs/1711.11585, 2017.
19. *Kingma D.P. and Ba J. Adam*: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
20. *Ulyanov D., Vedaldi A. and Lempitsky V.* Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022, 2016.