# An Intelligent System for Content Generation

Evgeny I. Nikolaev
notdeveloper@gmail.com

Pavel V. Dvoryaninov
paveldv92@mail.ru
Yaroslav Y. Lensky
yalenskiy@yandex.ru

Nikita S. Drozdovsky
nikitadrozdovsk@mail.ru

The North-Caucasus Federal University,
Institute of Information Technologies and Telecommunications,
Stavropol, Russian Federation

## Abstract

Generation graphical content from a single template image file in art manner is an important but difficult task for artificial neural networks, mostly due to the huge difference between classification existing data and producing new data. Nevertheless artificial intelligent systems capable of generating new content are important scientific task, because their working principles are close to the human thinking processes. Here we introduce an artificial system based on a Deep Neural Network that creates images, audio or 3D-content. In this work, we propose an content generative system for producing content by using pre-trained Deep Neural Network. This is made possible mainly two technical innovations. First, we propose to use different pre-trained neural networks, so that generative system can use optimized network parameters to produce new images. Content generative system and core Deep Neural Network are weakly bound components and we can obtain different system output by core replacement. Second, proposed artificial system can be used not only for image generation, but also for producing audio content and generation 3D-models with target style.

## 1 Introduction

The class of Deep Neural Networks [Kri12] that are most comprehensive in image processing problems are called Convolutional Neural Networks (CNNs). Convolutional Neural Networks are widely used in various visual recognition problems such as image classification [Kri12], object detection [Gir14], semantic segmentation [Mos15], visual tracking [Hon15], and action recognition [Sim14]. The representation power of CNNs leads to successful results; a combination of feature descriptors extracted from CNNs and simple off-the-shelf classifiers works very well in practice. Encouraged by the success in classification problems, researchers start to apply CNNs to more intelligent problems such as content generation.

Producing a new 3D-objects, video, audio content as well as images are extremely complex tasks. All that problems are close to human thinking process. And content generation is close to process of new information and knowledge creation by human. In art, especially painting, music and sculpture, humans have mastered the skill to

In: S. Hölldobler, A. Malikov, C. Wernhard (eds.): *YSIP2 – Proceedings of the Second Young Scientist's International Workshop on Trends in Information Processing, Dombai, Russian Federation, May 16–20, 2017*, published at http://ceur-ws.org.

create unique things through composing a complex interplay between the content and style of an image. Thus far the algorithmic basis of this process is unknown and there exists no artificial system with similar capabilities. But in area of visual perception such as face and object recognition near-human performance was demonstrated by a class of models called Deep Neural Networks. Here we propose a strategies for building artificial system based on a Deep Neural Networks that generates digital content. Proposed system can be used in game development, cloud computing, as a tool of the graphical editors, data augmentation and in many others.

We propose three architectures of the Intelligent System for Content Generation (ISCG):

1. ISCG based on single pre-trained CNN.
2. ISCG based on Deconvolutional Neural Network (DNN).
3. ISCG that contains cascade of several CNNs and DNNs.

## 2 The Core Component of the Intelligent System for Content Generation

Convolutional Neural Network consists of layers of computational units. Each layer of units can be understood as a collection of image filters, each of which extracts a certain feature from the input image. Thus, the output of a given layer consists of so-called feature maps: differently filtered versions of the input image. CNN is a core component of the proposed artificial system. We have to train CNN before using artificial generative system. Figure 1 shows the CNN structure used for classification teacher's face images in NCFU. The data for training process has been scrapped from official site of the NCFU.

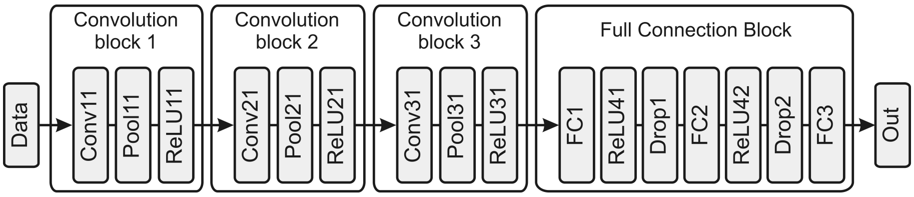

Figure 1: CNN structure

The CNN contains three convolutional blocks of layers. Each of them consists of three layers: convolutional layer (Conv11, Conv21, Conv31), pooling layer (Pool11, Pool21, Pool31) and nonlinear layers (ReLU1, ReLU2, ReLU3). The layer with ReLU type is a Rectified-Linear layer. There are also Input Data Layer (Data), full connection block and output block (Out). In described CNN we use max pooling layers. Data Layer realization depends on CNN functioning mode: training or deployment. During network learning process we use a special lmdb database to store images. But in the case of the use of deployment CNN version data layer presented by memory mapped input layer, therefore different applications can give the images on CNN input. Full connection block contains three Inner Product Payers (FC1, FC2, FC3), ReLU layers (ReLU41, ReLU42) and dropout layers (Drop1, Drop2). CNN output (Out) implementation also depends on network functioning mode. In learning mode output section of CNN consists of Softmax Layer and Accuracy Layer. Softmax Layer computes the multinomial logistic loss of the softmax of its inputs. Accuracy layer scores the output as the accuracy of output with respect to target. In case of deployment mode we use the only softmax layer as CNN output.

Convolutional Neural Network is a core component of the intelligent generative system. We have to train CNN before deployment. After learning procedure we have pretrined CNN: in fact we need only set of convolutional filter parameters. Figure 2 shows the CNN representation with filter parameters visualization (dimensions of layers and filters). Convolutional layers with dimensions are presented in Fig. 2.

Described CNN is realized on Caffe platform. Caffe is compiled with CuDNN library. Layers are defined in protobuf caffe file. We train CNN on NVidia GeForce Titan GPU.

## 3 ISCG Architectures

### 3.1 Direct Stylization Algorithm (DSA)

Generation a new image by using pretrained CNN from target style and input image is the simpliest way of the artificial generative system implementation [Gat15]. In that case we need only convolutional part of the pre-trained CNN and program which process all data. We use Python script to realize forward propagation through CNN.

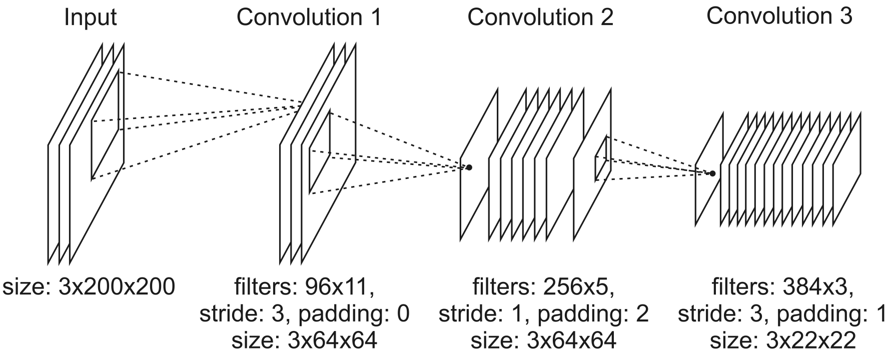

Figure 2: Dimensions of filters and layers

To obtain a stylization of an input image, we use a feature space designed to capture information from original training set. This feature space is built on top of the filter responses in each layer of the network. It consists of the correlations between the different filter responses over the spatial extent of the feature maps. By including the feature correlations of multiple layers, we obtain a stationary, multi-scale representation of the input image, which captures its texture information but not the global arrangement. The main idea of combining target style image and content input image is to minimize distance between CNN outputs for target image and content image. But we can use different methods for representation function of distance. One of the best approaches is to use Gramm matrix [Lem16].

A given input image is represented as a set of filtered images at each processing stage in the CNN. While the number of different filters increases along the processing hierarchy, the size of the filtered images is reduced by some downsampling mechanism leading to a decrease in the total number of units per layer of the network. We can visualise the the new content at different processing stages in the CNN by reconstructing the input image from only knowing the networks responses in a particular layer. We receive the styled image from from layers Conv11, Conv21, Conv31 (Fig. 1) of the core network. It's obviously that stylization from higher layer (Conv21, Conv31) is almost perfect. In higher layers of the network, detailed pixel information from the original input image is lost while the high-level content of the image is preserved.

Using DSA one can obtain a great variety of the new images. The key DSA advantages are:

1) DSA is the simplest method for generation styled images from target style and input image.

2) By using DSA one doesn't have to build and train another neural network such as deconvolutional generative neural network. For DSA implementation we can use the only CNN.

3) It's possible to replace core CNN in artificial generative system and obtain another output space of the styled images.

But for more complex tasks DSA is not suitable. We can't generate new audio or 3D content by using the only CNN. In addition DSA take a lot of time (2 minutes per image), therefore it's not suitable for stylization video and images in real time.

### 3.2 Generative Neural Network

Another model for ISCG is based on Generative Neural Network. This approach involves building additional Deconvolutional Neural Network which can generate images from the output of CNN [Noh15]. Figure 3 shows the architecture of the DNN based on CNN described above (Fig. 1). The main principle of DNN is to use layers which can complement input data. For that purpose Deconvolution Layers and Unpooling Layers are used.

We use unpooling layers in DNN, which perform the reverse operation of pooling and reconstruct the original size of input images. To implement the unpooling operation, we follow the similar method proposed in [Zei14]. It records the locations of maximum activations selected during pooling operation in switch variables, which are employed to place each activation back to its original pooled location. The output of an unpooling layer is an enlarged, yet sparse activation map. The deconvolution layers densify the sparse activations obtained by unpooling through convolution-like operations with multiple learned filters. However, contrary to convolutional layers, which connect multiple input activations within a filter window to a single activation, deconvolutional layers associate a single input activation with multiple outputs. The output of the deconvolutional layer is an

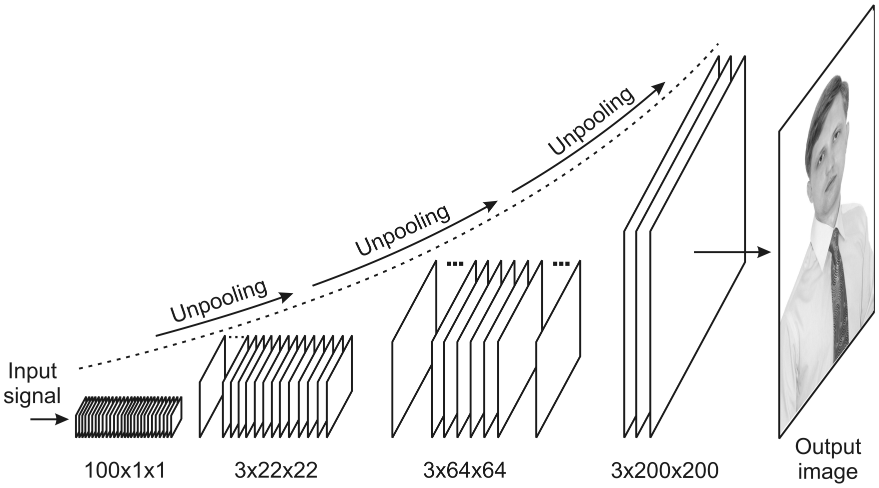

Figure 3: DNN layers

enlarged and dense activation map. We crop the boundary of the enlarged activation map to keep the size of the output map identical to the one from the preceding unpooling layer. The learned filters in deconvolutional layers correspond to bases to reconstruct shape of an input object. Similar to the convolution network, a structure of deconvolutional layers are used to capture different level of shape details. The filters in lower layers tend to capture overall shape of an object while the class-specific finedetails are encoded in the filters in higher layers.

Described DNN is not deep (just 5 layers), but contains a lot of associated parameters. The number of training examples for DNN is plenty: we can use CNN output signals to produce training set for DNN. But training DNN is still not trivial and takes a lot of time.

## 4 ISCG Applications

This section first describes our implementation details and experiment setup. Then, we analyse and evaluate the proposed artificial system in various aspects.

Image Stylization

The simplest way to generate new content is to implement ISCG with core CNN (DSA algorithm). The results of DSA are presented in Fig. 4.

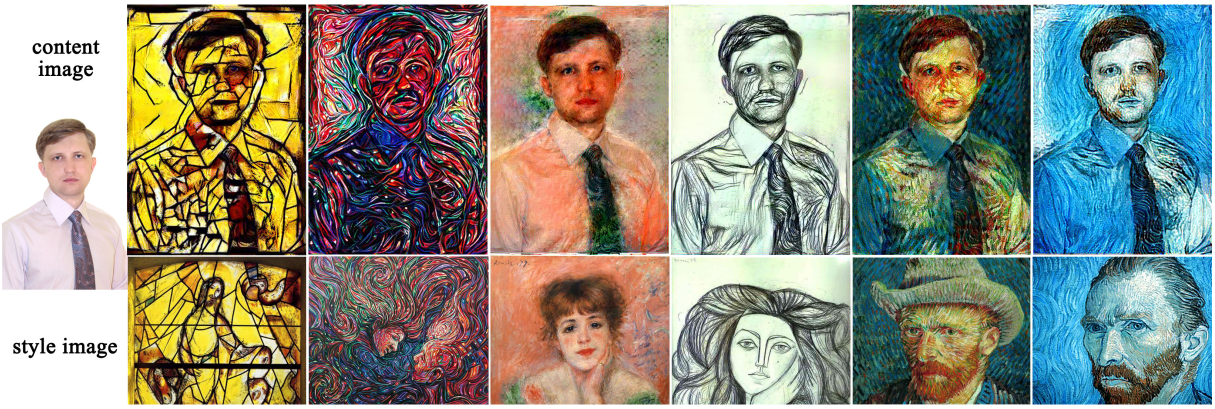

Figure 4: Content Generation by input image stylization

Each couple of the input images (style image and content image) can produce a set of the output styled images. We can receive output from different layers of the CNN. Also we can produce images with different stylization rate, which provided by number of iteration in DSA algorithm. Therefore, we can say about output space of images with dimensions $n$ (number of DSA iteration) and $l$ (CNN layer number).

### Texture Generation Task

Another way to generate new content is to use DSA with noise as content input image and texture as a style image. The main advantage of such approach is obtaining the output textures with different sizes (output texture size is more than size of the input style image). Figure 5 shows generated texture images from specified patterns.

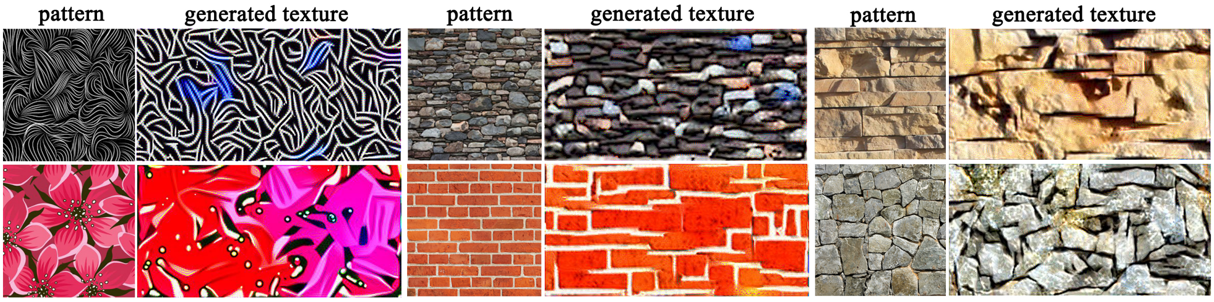

Figure 5: Texture Generation by noise image stylization

Obviously, most of the result images are wrong (output image doesn't present input pattern in right way). DSA can be used for texture generation in case when a simple patterns are used. The raeson of such results is that our CNN is not deep enough to extract apprópreate features. It's not possible to catch the high level image statistics by DSA algorithm. We can replace CNN core of the ISCG and use one of the public pre-trained CNN: vgg16, vgg19 [Zis14], googlenet [Sze14] or caffenet [Kri12].

### Smart Barcodes

The applications described above are based on DSA. For more complex tasks we need to use ISCG with DNN. One of the most interesting ISCG application envolves the coding/encoding information in intelligent manner. Today barcodes and QR-codes are widely used technologies. But there are several disadvantages in such technologies. The main of them is linked with aesthetical view of the barcode. Barcode is hard to memorize and get to know. Barcode generation approach is based on [Gri16]. The main idea involves the use several neural networks for barcode synthesis and digital code reconstruction. Figure 6 demonstrates that concept.

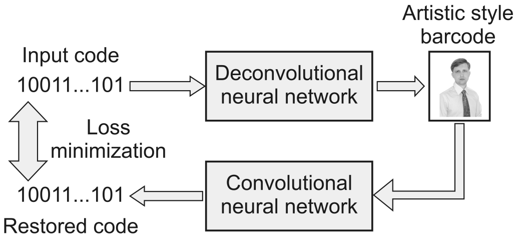

Figure 6: Smart Barcodes Generation Scheme

At the end of training process we will receive a system which can generate barcodes in artistic style. Recognition process for such barcodes is based on CNN.

### Mixed Content Generation

The most complex type of generative models is linked with Deep Generative Neural Networks (DGNNs). Similar approach is described in [Zei10, Zei11]. By using DGNN our ISCG can generate new graphical content. The key feature of this ISCG type is that intelligent system can generate new content that is not in training set. The model with DGNN is like a human art. In addition ISCG based on DGNN can produce not only images, but also audio and 3D content in a similar manner. Figure 7 illustrates the concept of the Mixed Content Generation procedure. By using such system it's possible to generate 3D-objects and audio content.

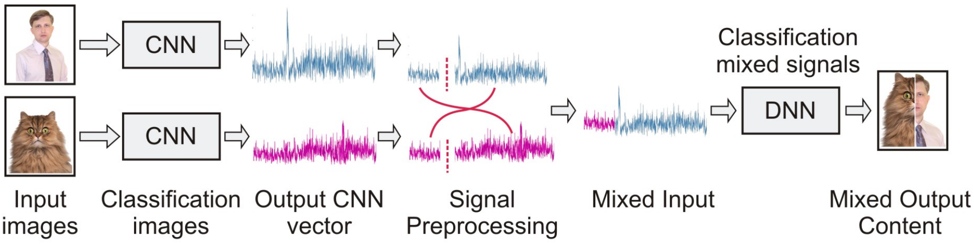

Figure 7: Mixed Content Generation Scheme

## 5 Conclusion

In this paper we demonstrated an application of artificial intelligent system based on deep neural network to the problem of content generation. Having tried different variants of the ISCG architecture, we showed that such architectures give different levels of accuracy and performance. In particular, ISCG with single CNN core (DSA algorithm) allows to extract simple features and to produce new styled content. ISCG with DNN core generates much more complex content in short time.

## References

[Kri12] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *In Proceedings of Neural Information Processing Systems (NIPS)*, 2012.

[Gir14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *In CVPR*, 2014.

[Mos15] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. *In CVPR*, 2015.

[Hon15] S. Hong, T. You, S. Kwak, and B. Han. Online tracking by learning discriminative saliency map with convolutional. *International Conference on Machine Learning (ICML)*, 2015.

[Sim14] K. Simonyan, A. Zisserman. Two-stream convolutional networks for action recognition in videos. *In NIPS*, 2014.

[Lem16] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. *International Conference on Machine Learning (ICML), New York*, 2016.

[Gat15] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. A Neural Algorithm of Artistic Style. *https://arxiv.org/abs/1508.06576*, 2015.

[Noh15] H. Noh, S. Hong, B. Han. Learning Deconvolution Network for Semantic Segmentation. *https://arxiv.org/abs/1505.04366*, 2015.

[Zei11] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive Deconvolutional Networks for Mid and High Level Feature Learning. *International Conference on Computer Vision (ICCV)*, 2011.

[Zei10] M. D. Zeiler, D. Kirshnan, G. W. Taylor, and R. Fergus. Deconvolutional Networks. *Computer Vision and Pattern Recognition*, 2010.

[Zis14] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ILSVRC*, 2014.

[Sze14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. Going Deeper with Convolutions. *ILSVRC*, 2014.

[Zei14] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *European Conference on Computer Vision (ECCV)*, 2014.

[Gri16] O. Grinchuk, V. Lebedev, and V. Lempitsky. Learnable Visual Markers. *Advances in Neural Information Processing Systems (NIPS), Barcelona*, 2016.