Automated Design of Accurate and Robust Image Classifiers with Brain Programming

Gerardo Ibarra-Vazquez
Facultad de Ingeniería, UASLP
gerardo.ibarra.v@gmail.com

Gustavo Olague
EvoVisión Laboratory, CICESE.
olague@cicese.mx

Cesar Puente
Facultad de Ingeniería, UASLP
cesar.puente@uaslp.mx

Mariana Chan-Ley
EvoVisión Laboratory, CICESE.
mchan@cicese.edu.mx

Carlos Soubervielle-Montalvo
Facultad de Ingeniería, UASLP
carlos.soubervielle@uaslp.mx

ABSTRACT

Foster the mechanical design of artificial vision requires a delicate balance between high-level analytical methods and the discovery through metaheuristics of near-optimal functions working towards complex visual problems. Evolutionary computation and swarm intelligence have developed strategies that automatically design meaningful deep convolutional neural network architectures to create better image classifiers. However, these architectures have not surpassed hand-craft models working with outdated problems with datasets of icon images. Nowadays, recent concerns about deep convolutional neural networks to adversarial attacks in the form of modifications to the input image can manipulate their output to make them untrustworthy. Brain programming is a hyper-heuristic whose aim is to work at a higher level of abstraction to develop automatically artificial visual cortex algorithms for a problem domain like image classification. This work's primary goal is to employ brain programming to design an artificial visual cortex to produce accurate and robust image classifiers in two problems. We analyze the final models designed by brain programming with the assumption of fooling the system using two adversarial attacks. In both experiments, brain programming constructed artificial brain models capable of competing with hand-crafted deep convolutional neural networks without any influence in the predictions when an adversarial attack is present.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**; **Machine learning**; **Computer vision**; **Object recognition**;

KEYWORDS

Secure, Face Recognition, Art Media Categorization, Adversarial Attacks, Convolutional Neural Networks, Brain Programming.

ACM Reference Format:
Gerardo Ibarra-Vazquez, Gustavo Olague, Cesar Puente, Mariana Chan-Ley, and Carlos Soubervielle-Montalvo. 2021. Automated Design of Accurate and Robust Image Classifiers with Brain Programming. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3449726.3463179

1 INTRODUCTION

Deep Learning with Deep Convolutional Neural Networks (DCNN) has been among the most popular and successful methods for solving image classification problems [4, 34]. Their architecture permits finding the best model parameters that fit the data for solving image classification problems. Evolutionary Computation (EC) and Swarm Intelligence (SI) have mainly contributed in two manners: 1) optimizing feature selection and 2) optimizing DCNN architectures. In this manner, researchers from EC and SI have developed strategies that automatically design meaningful DCNN architectures to develop new structures that improve their performance [3]. Recent approaches summarized in [16], have also explored hybridization of swarm and evolutionary computation algorithms by aggregating hyper-parameters' optimization during training.

Automatically design image classifiers have dealt with the extensive computational resources needed in the optimization process [3]. Also, the vast search space usually is constrained based on problem knowledge to reduce the exploration. So, it is crucial to pursue novel ways to generate better image classifiers to a given instance of the problem to produce solutions that solve more generic problems. However, DCCN architectures discovered by EC and SI have not exceeded hand-craft models, and they are still working with outdated problems with classical datasets [11, 25, 26]. Moreover, recent security concerns about the vulnerability of DCNN to Adversarial Attacks (AA) in slight modifications to the input image almost invisible to human vision make their predictions untrustworthy. Nevertheless, despite significant efforts to solve this problem, attacks have become more complex and challenging to defend [1].

Conversely to DCNN are brain-inspired computational models of the visual system, which pursue the imitation/understanding of the visual information processing that occurs in the brain. In brain-inspired computing, some authors studying the visual cortex refers to the natural process that occurs along the visual pathway according to the brain's neurological ventral-dorsal model [5] and the feature integration theory [29] to describe the process of visual

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463179

information to classify an image. The Artificial Visual Cortex (AVC) has a defined hierarchical structure inspired by the human visual cortex, thus simplifying the initial search space while using a set of atomic functions to extract the images' discriminant characteristics [18]. It is a flexible algorithmic framework that generates specific classification models due to the balance between the fix template method defining invariant parts of the program which are not subject to variation and the parts which can be adapted. The whole algorithm configuration provides the seeds to start the design from components of existing algorithms rather than search for algorithms with randomly initialized programs. This methodology leverage the designer to create computational models of image classification while using genetic programming as the search mechanism based on elementary functions and a hierarchical template design [8].

Brain programming (BP) is a hyper-heuristic capable of automatically design new algorithms by configuring functions to optimize complex models of the AVC by adjusting the operations within this intricate structure. Also, it uses the concept of composition of functions to extract features from images. Thus, BP differs from the data-driven models using a function-driven approach to extract and combine the relevant information to produce solutions that solve more generic problems. This metaheuristic approach proposes an evolutionary framework based on a template design that balances analysis and synthesis for developing automated image classifiers. The analytical part resides on the template where high-level processes are defined to secure concepts like invariance and hierarchical visual processing. The synthetic part is charged with heuristic discoveries through the application of genetic programming as the search method.

We employed BP to automatically improve the AVC structure to design accurate and robust image classifiers. The main goal of this work is to demonstrate the robustness of the AVC models' solutions applicable in two different image classification problems (Face Recognition and Art Media Categorization). We analyzed the behavior of the AVCs designed by BP with the assumption of an attempt to fool the system using AA, and we compared the performance and the vulnerability with a hand-crafted state-of-the-art DCNN (ResNet). The comparison is made because the automatically designed DCNN architectures are still not suitable for image classification problems such as Face Recognition and Art Media Categorization.

This paper is organized as follows. First, we outline the related work briefly to highlight the key ideas. Next, we present the security problem in deep learning, and then we outline the brain programming methodology. Later, we introduce two adversarial attacks and two different image classification problems. Next, we test the accuracy and robustness of image classifiers designed with BP, which has not been explored in the state-of-the-art evolutionary algorithms (EAs). Thus, this work intends to highlight the differences between such methods, opening the possibility to use EAs in the image classification pipeline to secure the predictions against adversarial attacks. Finally, we finish the paper with some conclusions and some ideas for future work.

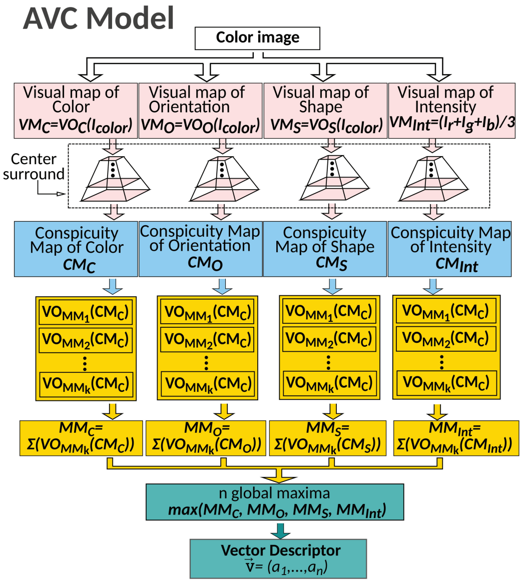
AVC Model
Color image
Visual map of Color $VM_C=VO_C(I_{color})$
Visual map of Orientation $VM_O=VO_O(I_{color})$
Visual map of Shape $VM_S=VO_S(I_{color})$
Visual map of Intensity $VM_{Int}=(I_r+I_g+I_b)/3$
Center surround
Conspicuity Map of Color CM_C
Conspicuity Map of Orientation CM_O
Conspicuity Map of Shape CM_S
Conspicuity Map of Intensity CM_{Int}
$VO_{MM_1}(CM_C)$
$VO_{MM_2}(CM_C)$
$VO_{MM_k}(CM_C)$
$MM_C=\Sigma(VO_{MM_k}(CM_C))$
$MM_O=\Sigma(VO_{MM_k}(CM_O))$
$MM_S=\Sigma(VO_{MM_k}(CM_S))$
$MM_{Int}=\Sigma(VO_{MM_k}(CM_{Int}))$
n global maxima
$max(MM_C, MM_O, MM_S, MM_{Int})$
Vector Descriptor
$\vec{v}=(a_1,...,a_n)$

Figure 1: Illustration of the AVC model.

2 RELATED WORK

The algorithms' mechanical design aims to automatically develop faster/better metaheuristics to a problem domain like image classification. The search for solutions usually starts from scratch, increasing the computational cost. The idea to alleviate this issue is to advance the design by providing existing components of algorithms or a complete algorithm structure–template method–to adapt/discover novel heuristics. EC and SI have developed strategies to automatically design DCNN architectures for image classification [3]. In [26], the authors use genetic algorithms for evolving DCNN architectures and connection values to address image classification problems. Authors from [11] employed a particle swarm optimization algorithm capable of automatically designing DCNN architectures for image classification with fast convergence. In [25], it is proposed to automatically evolve DCNN architectures by using a genetic algorithm based on ResNet[7] and DenseNet[9] blocks. However, these works are still working with outdated problems with classical datasets such as MNIST, MNIST-Fashion, CIFAR10, and CIFAR100, among other datasets with icon images. They have fallen short to be on par with hand-craft DCNN architectures.

The primary trend is to focus the design strategy on hand-crafted DCNN as a template method adapting it to a classification problem using transfer learning [28, 30]. For example, DCNN has achieved exemplary results in popular databases for face recognition, even some of them were pretrained for generic object recognition [7, 24, 27]. Also, pre-train DCNN models (AlexNet, VGG, GoogLeNet, ResNet, DenseNet) have been used to recognize basic artistic media from artworks [32]. They obtained comparable results with that of trained humans.

However, despite the progress made on evolutionary computation to build better image classifiers, an attempt to fool the system has not been considered in the mechanical design of image classification algorithms even though AA is a severe threat to DCNN

[1, 14, 31]. Face recognition systems demand reliability and security in their predictions because they are one of the most popular biometric used for person recognition because of their contactless and non-invasive procedure[33]. In museums and galleries, there are critical areas such as artist identification and forgery detection, where the confidence of the prediction must not depend on a system that an imperceptible perturbation can manipulate. This catastrophic scenario could lead to forgeries circulating on the market or be misattributed to a specific artist [20].

3 PROBLEM DESCRIPTION

Adversarial attacks usually are established as constraint optimization problems. For example, let $\mathbf{x}$ be the input image which is classified as $f(\mathbf{x})$, where f is the target image classifier. The objective is to find a perturbation ϵ such that $f(\mathbf{x}+\epsilon)$ predicts $y_t \neq y_{original}$. The perturbation ϵ is limited to be as imperceptible as possible with maximum modification constraint L measured by the length of vector ϵ. For targeted attack, y_t is an specified target class, and for non targeted attack, y_t is not specified. Therefore, targeted attacks find an optimal solution $\epsilon*$ for the following equation:

$$\begin{aligned} \min_{\epsilon*} \quad & J(f(\mathbf{x}+\epsilon), y_t) \\ \text{s.t.} \quad & ||\epsilon|| \leq L \quad . \end{aligned} \tag{1}$$

It minimizes the cost function J over the target class y_t. In a non-targeted attack, the goal is to find a perturbation $\epsilon*$ that maximizes the cost function's values J over the original predicted class $y_{original}$. That means to minimize the probability of the class $y_{original}$, and the optimization is defined as follows:

$$\begin{aligned} \max_{\epsilon*} \quad & J(f(\mathbf{x}+\epsilon), y_{original}) \\ \text{s.t.} \quad & ||\epsilon|| \leq L \quad . \end{aligned} \tag{2}$$

These attacks pose a serious threat to image classification security using deep neural networks. A simple description of this area's problem is that the data resolution is limited to 1/255 as most digital images use the 8-bit per channel. Then, if every element of a perturbation ϵ is smaller than the data resolution, the linear model will predict different an input $\mathbf{x}$ than to an adversarial example $\mathbf{x}_\epsilon = \mathbf{x} + \epsilon$.

It is expected for the classifier to predict $\mathbf{x}$ and $\mathbf{x}_\epsilon$ as the same class meanwhile $||\epsilon|| \leq L$, where L is too small to be discarded. However, if we consider the dot product of the neural network weights $\mathbf{w} \in \mathbf{R}^{M \times N}$ with the adversarial example $\mathbf{w}^\intercal \mathbf{x}_\epsilon = \mathbf{w}^\intercal \mathbf{x} + \mathbf{w}^\intercal \epsilon$, the activation will grow by $\mathbf{w}^\intercal \epsilon$. Thus, the perturbation can be imperceptible in the input but it obtain big changes to the output because the activation can grow linearly with n for ϵ.

The structure of DCNN constructs a mapping that uses pixels to analyze the image's content and assign the class in the data $(y, \mathbf{x})$ given by the dataset, which behaves extremely linear to be unaffected to adversarial examples. This mapping has a particular form of nested functions to make a deep structure that does the hard work of computing complicated math to find patterns throughout the image pixels; each nested function is called a layer. We outline an example of these structures on the following equation

$$y = f_{DCNN}(x) = f_3(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x}))) \quad , \tag{3}$$

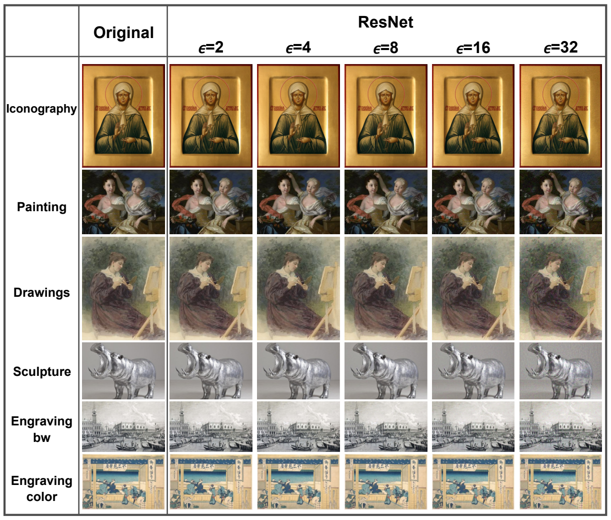

Figure 2: These images illustrate adversarial examples computed with the FGSM. The first column shows clean images from each class, and the subsequent columns show illustrations from adversarial examples using ResNet with a scale factor of $\epsilon = 2, 4, 8, 16, 32$.

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are vector functions of the following form:

$$\mathbf{f}_l(\mathbf{z}) = \mathbf{g}_l(\mathbf{W}_l\mathbf{z} + \mathbf{b}_l) \quad ,$$

with l denoting the index of the layer. $\mathbf{g}_l$ is the activation function, and the model parameters of $\mathbf{W}_l$ the weights and $\mathbf{b}_l$ the bias vector of the l layer. DCNN find the best model parameters for all the layers $\mathbf{w}$ that fits the data $\mathbf{x}$ to the label y, nonetheless every perturbation, as easy or difficult as it is to compute it affects the network. Additionally, adversarial examples often affect several models, whether the models have different architectures or are training with different datasets. They only need to be set up for the same task to be affected.

4 BRAIN PROGRAMMING

BP is a hyper-heuristic capable of automatically design new algorithms using the AVC model (see Figure 1) for solving computer vision problems [8, 18, 19]. This methodology extracts characteristics from images through a hierarchical structure inspired by the brain's functioning. BP proposes a multi-tree representation for individuals. The main goal is to obtain a set of evolutionary visual operators ($EVOs$), also called visual operators (VOs), embedded within the hierarchical structure of the AVC. This section briefly described BP, but further details are explained in [8, 18, 19].

4.1 Data Modeling in Brain Programming

BP proposes to solve the image classification problem from the standpoint of data modeling through Genetic Programming. BP fits the data by finding elemental functions that perform a task instead of conventional approaches to finding the best-fit parameters. In this manner, BP defines the solution to the image classification problem as follows:

Table 1: Functions and terminal list for the VO.

Dimension	Functions	Description	Terminals	Description
VO_O	$A+B$, $A-B$, $A\times B$, A/B, $k+A$, $k-A$, $k\times A$, A/k, $\lvert A\rvert$, $\lvert A+B\rvert$, $\lvert A-B\rvert$, $log(A)$, $(A)^2$, $\sqrt{A}$, $round(A)$, $\lfloor A\rfloor$, $\lceil A\rceil$, $inf(A,B)$, $sup(A,B)$, $G_{\sigma=1}(A)$, $G_{\sigma=2}(A)$, $D_x(A)$, $D_y(A)$, $thr(A)$	Arithmetic functions between images or constants k, absolute values, transcendental functions, square, square root, rounding, infimum, supremum, convolution with a Gaussian filter, derivatives, and threshold applied to images A and/or B	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, $D_x(I_x)$, $D_{xx}(I_x)$, $D_y(I_x)$, $D_{yy}(I_x)$, $D_{xy}(I_x)$	Elements of I_{color} and its derivatives
VO_C	$A+B$, $A-B$, $A\times B$, A/B, $log(A)$, $exp(A)$, $(A)^2$, $\sqrt{A}$, $(A)^c$, $thr(A)$	Arithmetic and transcendental functions, square, square root, image complement, rounding and threshold applied to images A and/or B	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, $Op_{r-g}(I)$,$Op_{b-y}(I)$	Elements of I_{color} and color opponencies: red-green and blue-yellow
VO_S	$A+B$, $A-B$, $A\times B$, A/B, $k+A$, $k-A$, $k\times A$, A/k, $round(A)$, $\lfloor A\rfloor$, $\lceil A\rceil$, $thr(A)$, $A\oplus SE_d$, $A\oplus SE_s$, $A\oplus SE_{dm}$, $A\ominus SE_d$, $A\ominus SE_s$, $A\ominus SE_{dm}$, $A\circledcirc SE_s$, $A\odot SE_s$, $Sk(A)$, $Perim(A)$, $A\circledast SE_d$, $A\circledast SE_s$, $A\circledast SE_{dm}$, $T_{hat}(A)$, $B_{hat}(A)$	Arithmetic functions between images or constants k, rounding, threshold, and morphological operators: dilation,erosion, open, close with disk, square, and diamond structural element; skeleton, hit or miss, bottom-hat, top-hat	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v	Elements of I_{color}
VO_{MM}	$A+B$, $A-B$, $A\times B$, A/B, $\lvert A+B\rvert$, $\lvert A-B\rvert$, $log(A)$, $(A)^2$, $\sqrt{A}$, $G_{\sigma=1}(A)$, $G_{\sigma=2}(A)$, $D_x(A)$, $D_y(A)$	Arithmetic functions between images or constants k, absolute values, transcendental functions, square, square root, convolution with a Gaussian filter, and derivatives	CM_d, $D_x(CM_d)$, $D_{xx}(CM_d)$, $D_y(CM_d)$, $D_{yy}(CM_d)$, $D_{xy}(CM_d)$	Conspicuity Maps and its derivatives

$$y = min(f(\mathbf{x}, \mathbf{F}, \mathbf{T}, \mathbf{a})) \quad , \tag{4}$$

where $(y, \mathbf{x})$ are the label and the image, respectively, given by the dataset. $\mathbf{F}$ represents the set of functions. $\mathbf{T}$ defines the terminal set, and $\mathbf{a}$ are the parameters controlling the algorithm. Thus, the technique requires two things: (1) a method of feature extraction and (2) a suitable criterion $\mathbf{Q}$ for the minimization.

BP is the algorithm for looking at an optimal feature extraction from the images for each visual operator embedded into the AVC by tuning $(\mathbf{F}, \mathbf{T}, \mathbf{a})$. In order to set up BP to image classification tasks, the criterion for the minimization $\mathbf{Q}$ uses a support vector machine (SVM) to learn a mapping $f(\mathbf{x})$ that associates the image representation $\mathbf{x_i}$ to labels y_i. BP outlines the recognition problem in terms of a binary classification to find a decision boundary that best separates the class elements.

Table 2: Parameters applied in the BP algorithm.

Parameters	Description
Generations	30
Initial Population	30
Crossover at chromosome level	0.4
Crossover at gene level	0.4
Mutation at chromosome level	0.1
Mutation at gene level	0.1
Tree depth	Dynamic depth selection
Dynamic max depth	7 levels
Real max depth	9 levels
Selection	Tournament with lexicographic parsimony pressure
Survival	Elitism

4.2 Evolving an Artificial Visual Cortex

BP uses an evolutionary paradigm to evolve a population of individuals represented by the AVC template (see Figure 1). Each contains a set of syntactic trees defining the VOs that constructs the AVC structure to extract features from color images. This procedure gets a descriptor vector that encodes salient characteristics from the image. Then, we apply an SVM to calculate the classification accuracy for a given training image database to obtain individual fitness.

4.3 Structure Representation

Individuals within the population contain a variable number of syntactic trees, ranging from 4 to 12, one for each evolutionary visual operator (VO_O, VO_C, VO_S, VO_I) regarding orientation, color, shape, and intensity; and at least one tree to merge the visual maps produced and generated with the Mental Maps (MM), see Table 1. All atomic functions within each VO are defined according to expert knowledge to highlight characteristics related to the respective feature dimension and updated through genetic operations. Details about these visual operators' usage are explained in detail in [2, 8, 18].

4.4 Visual Maps

Each input image is transformed to build the set $I_{color} = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$, where each element corresponds to the color components of the RGB (red, green, blue), CMYK (Cyan, Magenta, Yellow, and black) and HSV (Hue, Saturation, and Value) color spaces. Elements on I_{color} are the inputs to four VOs defined by each individual. It is important to note that each solution in the population should be understood as a complete system and not only as a list of tree-based programs. Individuals represent a possible configuration for feature extraction that describes input images and are optimized through the evolutionary process. Each VO is a function applied to the input image to extract specific features from it, along with information streams of color, orientation, shape, and intensity; each of these properties is called a dimension. The output to VO is an image called Visual Map (VM) for each dimension.

4.5 Conspicuity Maps

The following process is the center-surround process; it efficiently combines the information from the VMs and helps detect scale invariance in each of the dimensions. This process is performed by applying a Gaussian smoothing over the VM at nine scales; this processing reduces the visual map's size by half on each level forming a pyramid. Subsequently, the six levels of the pyramid are extracted and combined. Since the levels have different sizes, each level is normalized and scaled to the visual map's dimension using polynomial interpolation. This technique emulates the center-surround process of the biological system. After extracting features, the brain receives stimuli from the vision center and compares it with the receptive field's surrounding information. The goal is to process the images so that the results are independent of scale changes. The entire process ensures that the image regions are responding to the indicated area. This process is carried out for each characteristic dimension; the results are called Conspicuity Maps (CM), focusing only on the searched object by highlighting the most salient features.

4.6 Mental Maps

After obtaining the most saliency features, the next stage along the AVC is to compute the Mental Maps (MMs) to define a descriptor vector used as input to a classifier for categorization purposes. The information from CMs is synthesized to build the set of MMs, which discriminates unwanted information.

The AVC model uses a function set to extract the images' discriminant characteristics; it uses a functional approach. Thus, a set of k VOs is applied to the CMs for the construction of the MMs. These VOs correspond to the remaining part of the individual that has not been used. Unlike the operators used for the VMs, the operators' whole set is the same for all the dimensions. These operators filter the visual information and extract the information that characterizes the object of interest. Equation (5) computes the MMs for each dimension, where d is the dimension, and j represents the set VO_{MM_j} cardinality.

$$MM_d = \sum_{i=1}^{j} VO_{MM_i}(CM_d). \tag{5}$$

4.7 Fitness Function

The next step in the model is constructing the image descriptor vector (DV), where the method concatenates the four MMs and uses a max operation to extract the n highest values. Once it is obtained the descriptor vectors from all the images in the database, the method trains an SVM. The classification score obtained by the SVM indicates the fitness of the individual.

4.8 Initialization, Evolutionary Process, and Solution Designation

Once it is defined the AVC structure from each individual, the parameters of the evolutionary process of BP are set as Table 2 and evaluated with the image database. A random initial population is then created using a ramped half-and-half technique, which selects half of the individuals with the grow method and half with the full method. After the first generation is evaluated, individuals are selected from the population with a probability based on fitness to participate in the genetic recombination, and the best individual is retained for further processing. The new individual of the population is created from the selected individual by applying genetic operators. Like genetic algorithms, BP executes the crossover between two selected parents at the chromosome level using a "cut-and-splice" crossover. Thus, all data beyond the selected crossover point is swapped between both parents A and B. The result of applying a crossover at the gene level is performed by randomly selecting two subtree crossover points between both parents. The selected genes are swapped with the corresponding subtree in the other parent. The chromosome level mutation leads to selecting a given parent's random gene to replace such substructure with a new randomly mutated gene. The mutation at the gene level is calculated by applying a subtree mutation to a probabilistically selected gene; the subtree after that point is removed and replaced with a new subtree.

Finally, the evolutionary process is terminated until one of these two conditions is reached: (1) an acceptable classification rate, or (2) the total number of generations. Thus, the evolutionary process reaches an optimum population that contains the best solution to the problem.

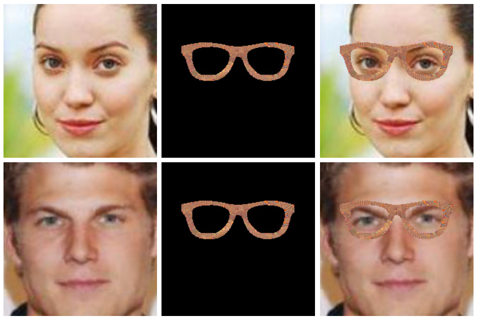

Figure 3: These images illustrate adversarial examples computed with the facial accessories perturbations. The first column shows the clean face images, the second column presents the precomputed ResNet glasses frame, and the third column shows the resulting adversarial examples.

Table 3: Dataset construction from CelebA images.

Class	training	validation	testing
Faces	1000	300	800
Background	1000	300	800

4.9 Hands-on Artificial Evolution

The use of random principles is overused in evolutionary computation. Thus, Olague and Chan-Ley adapted a methodology to avoid the unnecessary application of arbitrary or unplanned solutions within an algorithm to advance towards a more goal-oriented methodology [17]. It is not feasible to leave a methodology to discover the best solution when they have complex structures, but helping it with previous discoveries will guide the search in a better direction. Hence, the idea was to use the best solutions discovered

Table 4: Total number of images per class obtained from Kaggle and Wikiart Databases

	Drawings	Engraving gray scale	Engraving color	Painting	Iconography	Sculpture	Caltech Background
Train	553	426	30	1021	1038	868	233
Validation	553	284	19	1021	1038	868	233
Wikiart	204	695	1167	2089	251	116	233
Wikiart Landscapes				136			

during previous searches as the initial population to set a new experiment to find a better solution.

Brain programming is a highly demanding computational paradigm. A balance should be found to create programs that can solve non-trivial problems within the state-of-the-art in a reasonable amount of time. The idea is to continue the evolution from the best local minimum discovered so far. The proposed technique significantly improves the performance of previous results. The idea of hands-on evolution works for computationally demanding problems. It is a simple strategy that saves computational time because this kind of results cannot be obtained by simply continuing the random initial population's approach.

5 EXPERIMENTS

Reliable predictions are a highly valuable characteristic regarding face recognition system development following security and confidence of the recognition. We propose the assumption of an attempt to fool the system and highlight the differences between the renowned DCNN who has performed well in object and face recognition (ResNet [7]) due to automatically designed DCNN architectures are still not suitable for image classification problems beyond icon images, and an Evolutionary Paradigm (BP) who has obtained comparable results with AlexNet[12]. We considering performance and security against adversarial attacks in the most straightforward face recognition and artwork classification experiments. We employed classification accuracy as a measure of performance for the classifiers, which is simply the rate of correct classifications given by the following formula:

$$Accuracy = \frac{1}{N}\sum_{n=1}^{N} d(y'_n, y_n), \tag{6}$$

where N is the total of test images, y'_n is the predicted label for the image n, y_n is the original label for the image n, and $d(x, y) = 1$ if $x = y$ and 0 otherwise. We consider the training, validation, and testing stages. The aim is to emulate a real-world scenario where the proposed models consider the standard benchmark procedures. Additionally, we analyze two AA threats to evade recognition. Further experimentation about this study can be found in [10].

5.1 Adversarial Attacks

Adversarial attacks are classified depending on the model's available information and the desired attack to predict a specific class. Hence, we choose two untargeted attacks: the Fast Gradient Sign Method, which is the most widely used method for computing adversarial examples given an input image due to its easy implementation, and the facial accessories perturbations, which are physically realizable and inconspicuous pair of eyeglass frames to evade recognition. We briefly introduce them in the following paragraphs.

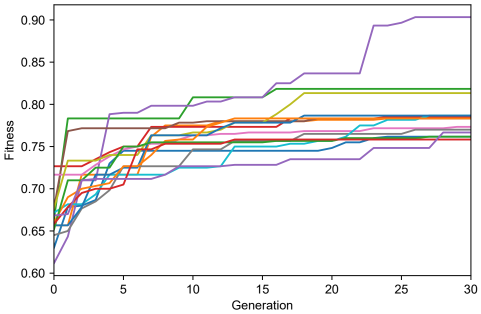

Figure 4: Fitness evolution progress for the best solution during the validation stage of BP. Each plot represents one of the 15 runs.

5.1.1 Fast Gradient Sign Method. The FGSM introduced in [6], proposes to increase the loss of the classifier by solving the following equation: $\rho = \epsilon \text{ sign}(\nabla J(\theta, \mathbf{x}, y_l))$, where $\nabla J()$ computes the gradient of the cost function around the current value of the model parameters θ with the respect to the image $\mathbf{x}$ and the target label y_l. sign() denotes the sign function, which ensures that the magnitude of the loss is maximized and ϵ is a small scalar value that restricts the norm L_∞ of the perturbation.

The perturbations generated by FGSM take advantage of the linearity of DCNN in the higher dimensional space to make the model misclassify the image. The implication of the DCNN' linearity discovered by FSGM is that exists transferability between models. Authors in [13] reported that with the ImageNet dataset, the top-1 error rate using the perturbations generated by FGSM is around 63-69% for $\epsilon \in [2, 32]$. Figure 2 shows example images from the FGSM.

5.1.2 Facial Accessories Perturbations. The Facial Accessories Perturbations infer that the attack is made with the targeted model's knowledge to evade the recognition [23]. In this manner, facial accessories are used to perform the attacks, which in this case are eyeglasses frames. The advantage of facial accessories is that they can be easily realizable in real-world conditions. Furthermore, eyeglasses are an everyday facial accessory that is natural for people to wear, helping the attacks be feasible.

Hence, a set of eyeglasses frames is employed to physically realize the attack, ensuring that the perturbation effectively misclassifies more than one image. In order to find a perturbation that performs the attack, the following optimization problem needs a solution:

$$\min_r \sum_{x \in X} -softmaxloss(f(x + r), c_x), \tag{7}$$

where the perturbation r would maximize the $softmaxloss(f(x + r), c_x)$ value to minimize the probability of the class c_x. To guarantee the generality of perturbations, we need to look for complex models that can cause any image in a set of inputs to be misclassified. Hence,

Table 5: Performance of the best individuals of BP in each run.

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Training	0.7175	0.7705	0.7810	0.7620	0.8925	0.7700	0.7165	0.7605	0.8420	0.7800	0.8065	0.7840	0.7195	0.7180	0.7500
Validation	0.7617	0.7833	0.8183	0.7850	0.9033	0.7833	0.7733	0.7700	0.8133	0.7867	0.7867	0.7833	0.7617	0.7583	0.7667

Table 6: Results obtained using the ResNet eyeglass attack. Each method presents its classification accuracy for training, validation, and testing, and the adversarial examples using the pre-computed glasses from ResNet. The third row present results of a two-sample Kolmogorov-Smirnov test between both methods.

	Clean Images			ResNet Eyeglasses		
	train	val	test	train	val	test
BP	94.1	95.67	93.19	94.3	94.33	95.75
ResNet	99.75	99.67	99.94	4.70	5.67	1.38
h	1	1	1	1	1	1

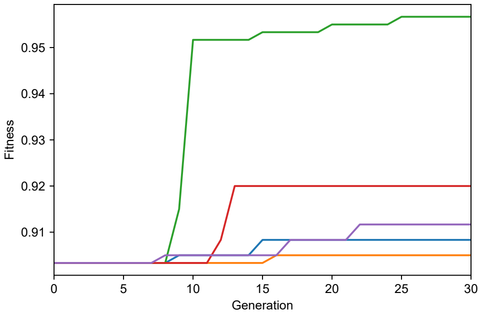

Figure 5: Fitness evolution progress for the best solution during the five runs of the hands-on artificial evolution.

the attack requires a set of images, X, and finds a single perturbation that optimizes her objective for every image $x \in X$. Figure 3 shows example images from the eyeglasses frame perturbation.

5.2 Dataset

We use a widely used face recognition dataset named CelebFaces Attributes (CelebA)[15]. It consists of a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover significant pose variations and background clutter. Additionally, it has enormous diversities, large quantities, and rich annotations, including 10,177 identities, 202,599 face images, five landmark locations, 40 binary attributes annotations per image.

As the experiment required a significant number of images, in Table 3, we provide the number of images for each set of images for training, validation, and testing stages. Hence, we randomly construct three sets of images from the CelebA dataset and manually fine-tune changing images to preserve the face diversity and front-facing images. Additionally, we use a Multi-task Cascaded Convolutional Networks (MTCNN) to perform face detection and alignment using the DeepFace library from [22]. We constructed the background class from landscape images from the ml5 project datasets [21].

Regarding the second problem, the training and validation set of images is obtained from the Kaggle website’s digitized artwork dataset. This dataset is composed of five categories of art media: drawing, painting, iconography, engraving, and sculpture. For the class engraving, there were two different kinds of engravings. Most of them were engravings with only one color defining the art piece. The other style was Japanese engravings, which introduce color to the images. So, the engraving class was split into engraving grayscale and engraving color. For testing, it is used a standard database WikiArt, where it was selected the images from the same categories. Since the Wikiart engraving class is grayscale, the ukiyo-e class (Japanese engravings) from Wikiart was used as the engraving color class. The set of images of the category landscapes painted by renowned artists is added to test the painting class. In Table 4 is referenced the number of images from each of the datasets.

5.3 Results

In this section, we present and discuss the experimental results of this work. Firstly, regarding the face recognition problem, we show in Figure 4 the fitness evolution progress of BP in the training phase. It is seen that most of the runs converged to approximately 75% of the validation accuracy, two runs achieved around 80%, and one run obtained approximately 90% (see Table 5). Hence, we can see that it is not easy to reach satisfactory solutions in the search space due to the complex structure of the AVC departing from random individuals, but it was possible to obtain an excellent individual. Nevertheless, due to the BP’s high computational cost, it was possible only to execute 15 runs with a mean execution time of 40.18 hours and a standard deviation of 1.04 hours in a server with an Intel Xeon Silver 4114 CPU and 32 Gb of RAM.

Therefore, we follow the hands-on artificial evolution strategy from [17] in which we selected the best two individuals from each run to construct an initial population to search for new individuals. Figure 5 shows the fitness evolution progress of five runs from this strategy. It can be seen the guide from the previous experiments delivered an increase of up to 5% in the performance. Hence, we validate the advantage of the hands-on evolution strategy to get out of local minima, thus helping the methodology to discover better solutions.

Next, we present in Table 6 the outcome of each model for the clean images, and when it is applied, the eyeglasses frame perturbation to the training, validation, and testing datasets. Each method presents its accuracy to each dataset. We observed that ResNet surpassed BP in all sets of clean images (training, validation, and testing). However, as we add the eyeglasses frame perturbation to all sets of face images, the AA’s effect becomes enormous. It is shown that ResNet completely drops its outstanding performance as the eyeglasses frame is present in the face images, making almost every face image in the three datasets evade the recognition.

Table 7: Results obtained after applying BP and ResNet on the Kaggle Dataset. Each method presents its classification accuracy for training, validation, and the adversarial examples using the FGSM computed from the validation dataset at $\epsilon = 2, 4, 8, 16, 32$.

	Brain Programming							ResNet						
	train	val	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	train	val	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$
Sculpture	93.19	93.26	**92.88**	**92.88**	**92.79**	**92.79**	**92.70**	**100**	**96.88**	84.88	84.88	45.92	25.30	19.07
Painting	99.68	**99.04**	**98.8**	**98.8**	**98.56**	**98.64**	**98.56**	**100**	97.85	86.92	86.92	43.94	31.82	40.75
Engraving gray scale	89.76	92.05	91.70	91.70	**92.23**	**92.05**	**91.53**	**100**	**100**	**95.58**	**95.58**	78.98	64.49	63.07
Engraving color	**98.33**	97.37	**97.37**	**97.37**	**97.37**	**97.37**	**97.37**	95.00	**100**	52.63	52.63	13.16	2.63	21.05
Iconography	92.84	91.42	**91.42**	**91.42**	**91.42**	**91.42**	**91.42**	**100**	**98.9**	90.24	90.24	52.01	29.03	32.10
Drawings	96.56	90.59	**90.59**	**90.59**	**90.59**	**90.59**	**90.59**	**99.87**	**94.44**	72.9	72.9	31.04	23.41	22.77

Table 8: Results obtained after applying BP and ResNet on the Wikiart Dataset. Each method presents its classification accuracy for testing and the adversarial examples from the FGSM computed from the test dataset at $\epsilon = 2, 4, 8, 16, 32$.

	Brain Programming						ResNet					
	test	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	test	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$
Sculpture	90.54	**90.83**	**90.83**	**90.83**	**90.83**	**90.83**	**92.63**	75.81	75.81	46.61	34.81	33.92
Painting	**100**	**95.65**	**95.65**	**95.65**	**95.65**	**95.65**	94.23	64.86	64.86	15.25	13.01	15.07
Painting Landscapes	**100**	**100**	**100**	**100**	**100**	**100**	95.12	72.36	72.36	42.82	33.60	38.48
Engraving gray scale	91.55	91.97	91.97	**91.80**	**91.97**	**91.80**	**99.83**	**93.22**	**93.22**	71.55	59.41	61.09
Engraving color	89.92	**89.68**	**89.68**	**89.74**	**89.98**	**89.80**	**96.40**	49.13	49.13	6.84	05.58	10.74
Iconography	91.74	**91.66**	**91.66**	**91.66**	**91.58**	**91.50**	**96.49**	76.86	76.86	38.64	25.21	21.49
Drawings	**94.05**	**93.81**	**93.81**	**93.59**	**93.59**	**94.05**	90.85	71.85	71.85	36.16	24.49	24.49

The powerful effect of the eyeglasses frame perturbation is seen on the training dataset, where even though ResNet has identified the faces images from the training stage, the perturbation makes them evade the recognition. Meanwhile, BP demonstrated to remain with a maximum variation of 2.56% of its original score, proving its security to recognize faces even the images are perturbed with an eyeglasses frame.

Additionally, Table 6 shows a two-sample Kolmogorov-Smirnov test between each pair of prediction confidences at each stage between DCNN and BP. h is one if the test rejects the null hypothesis at the 5% significance level and 0 otherwise. Hence, all the prediction confidences at each stage between both methods denoted significant differences by rejecting the null hypothesis.

Tables 7-8 present the results obtained for the artwork categorization problem using the FGSM. Table 7 shows that ResNet has surpassed BP in almost every class when it is testing using the validation dataset except for the painting. However, as we add the perturbations to the validation images, the effect of the adversarial attacks becomes more notable. It is shown how the performance of ResNet could deteriorate. In the worst case of the Engraving color images, there is a drop in performance from 97.37% to 2.63% of classification accuracy. On the other hand, BP preserves its performance on all experiments even when we added the strongest perturbation of $\epsilon = 32$. Hence, if we take a look at the best performances of each one of the comparisons (bold numbers), BP outperforms ResNet.

In the testing phase (see Table 8), we have that BP obtained notable better results for painting, painting landscapes and drawings. In contrast, ResNet obtained superior performance on engraving grayscale, engraving color, and iconography. Regarding the sculpture class, BP matches the performance of ResNet with a difference around 2.09%. Then again, the susceptibility from ResNet to adversarial examples is a big disadvantage. Its performances fall abruptly in all classes; meanwhile, the BP output remains steady.

6 CONCLUSIONS AND FUTURE WORK

Adversarial attacks are a severe threat to the security of deep convolutional neural networks in image classification. Their performance can be extremely manipulated with subtle perturbations generated by FGSM and a physically realizable and inconspicuous eyeglasses frame to evade recognition. However, the BP which is inspired by the brain's behavior demonstated to automatically design AVC models capable of competing with DCNN and safeguarded the predictions' integrity by remaining steady in its performance.

This work innovates by introducing the assumption of an attempt to fool the system and highlighting the differences between a renowned DCNN (ResNet) and the AVCs generated by BP. We also considered contrast performance and security against adversarial attacks in the most explicit face recognition and artwork classification problems. ResNet's performance was extremely weakened in both experiments, either with such small perturbations from FGSM and the facial accessories perturbation. In contrast, the AVC models from BP resist the attempt to mislead the system. Additionally, a two-sample Kolmogorov-Smirnov test confirmed that DCNN and the AVC models designed with BP are statistically different. These results open the possibility of using evolutionary computation in the face recognition and artwork classification pipeline to protect the predictions. Furthermore, we validate the hands-on strategy beyond a pure random initialization that helped get out from the local minima to discover better solutions.

Finally, future work from these results is considered by increasing the number of DCNN to verify such perturbations' transferability effect. Additionally, we want to study the behavior of mainstream computer vision approaches based on handcrafted features for face recognition and artwork classification problems.

ACKNOWLEDGMENTS

This research was founded by CICESE through Project 634-135, "Estudio de la programación cerebral en problemas de reconocimiento a gran escala y sus aplicaciones en el mundo real", and Universidad Autónoma de San Luis Potosí. The first author generously acknowledges the scholarship paid by the National Council for Science and Technology of Mexico (CONACyT) under Grant 469034-560085.

REFERENCES

[1] Naveed Akhtar and Ajmal Mian. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* 6 (2018), 14410–14430.

[2] Mariana Chan-Ley and Gustavo Olague. 2020. Categorization of digitized artworks by media with brain programming. *Applied Optics* 59 14 (2020), 4437–4447.

[3] A. Darwish, A. Hassanien, and Swagatam Das. 2019. A survey of swarm and evolutionary computing approaches for deep learning. *Artificial Intelligence Review* 53 (2019), 1767–1812.

[4] P. Druzhkov and V. Kustikova. 2016. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognition and Image Analysis* 26 (2016), 9–15.

[5] Melvyn A. Goodale and A.David Milner. 1992. Separate visual pathways for perception and action. *Trends in Neurosciences* 15, 1 (1992), 20 – 25. https://doi.org/10.1016/0166-2236(92)90344-8

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*. 11.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[8] Daniel E. Hernández, Eddie Clemente, Gustavo Olague, and José L. Briseño. 2016. Evolutionary multi-objective visual cortex for object classification in natural images. *Journal of Computational Science* 17 (2016), 216 – 233. https://doi.org/10.1016/j.jocs.2015.10.011

[9] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[10] Gerardo Ibarra-Vazquez, Gustavo Olague, Mariana Chan-Ley, Cesar Puente, and Carlos Soubervielle-Montalvo. 2021. Brain Programming is Immune to Adversarial Attacks: Towards Accurate and Robust Image Classification using Symbolic Learning. (2021). arXiv:cs.CV/2103.01359

[11] Francisco Erivaldo Fernandes Junior and G. Yen. 2019. Particle swarm optimization of deep neural networks architectures for image classification. *Swarm Evol. Comput.* 49 (2019), 62–74.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.

[13] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial Machine Learning at Scale. *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings* (2017), 17.

[14] Yuancheng Li and Yimeng Wang. 2018. Defense Against Adversarial Attacks in Deep Learning. *Applied Sciences* 9, 1 (Dec 2018), 76. https://doi.org/10.3390/app9010076

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[16] Takumi Nakane, B. Naranchimeg, Haitian Sun, Xuequan Lu, Takuya Akashi, and Chao Zhang. 2020. Application of evolutionary and swarm optimization in computer vision: a literature survey. *IPSJ Transactions on Computer Vision and Applications* 12 (2020), 1–34.

[17] Gustavo Olague and Mariana Chan-Ley. 2020. *Hands-on Artificial Evolution Through Brain Programming*. Genetic Programming Theory and Practice XVII, Springer International Publishing, Cham, 227–253. https://doi.org/10.1007/978-3-030-39958-0_12

[18] G. Olague, E. Clemente, D. E. Hernández, A. Barrera, M. Chan-Ley, and S. Bakshi. 2019. Artificial Visual Cortex and Random Search for Object Categorization. *IEEE Access* 7 (2019), 54054–54072.

[19] Gustavo Olague, Daniel E. Hernández, Paul Llamas, Eddie Clemente, and José L. Briseño. 2019. Brain programming as a new strategy to create visual routines for object tracking. *Multimedia Tools and Applications* 78, 5 (2019), 5881–5918.

[20] Gustavo Olague, Gerardo Ibarra-Vázquez, Mariana Chan-Ley, Cesar Puente, Carlos Soubervielle-Montalvo, and Axel Martinez. 2020. A Deep Genetic Programming Based Methodology for Art Media Classification Robust to Adversarial Perturbations. In *Advances in Visual Computing*, George Bebis, Zhaozheng Yin, Edward Kim, Jan Bender, Kartic Subr, Bum Chul Kwon, Jian Zhao, Denis Kalkofen, and George Baciu (Eds.). Springer International Publishing, Cham, 68–79.

[21] ML5 Project. 2020. Landscapes dataset. (2020). Retrieved 10/1/2020 from https://ml5js.org/

[22] Sefik Ilkin Serengil and Alper Ozpinar. 2020. LightFace: A Hybrid Deep Face Recognition Framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 23–27. https://doi.org/10.1109/ASYU50717.2020.9259802

[23] Mahmood Sharif, Sruti Bhagavatula, L. Bauer, and M. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016).

[24] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (2015), 14.

[25] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G. Yen. 2020. Completely Automated CNN Architecture Design Based on Blocks. *IEEE Transactions on Neural Networks and Learning Systems* 31, 4 (2020), 1242–1254. https://doi.org/10.1109/TNNLS.2019.2919608

[26] Y. Sun, B. Xue, M. Zhang, and G. G. Yen. 2020. Evolving Deep Convolutional Neural Networks for Image Classification. *IEEE Transactions on Evolutionary Computation* 24, 2 (2020), 394–407. https://doi.org/10.1109/TEVC.2019.2916183

[27] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1701–1708.

[28] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International conference on artificial neural networks*. Springer, 270–279.

[29] Anne M. Treisman and Garry Gelade. 1980. A feature-integration theory of attention. *Cognitive Psychology* 12 (1980), 97–136. https://doi.org/10.1016/0010-0285(80)90005-5

[30] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[31] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. 2020. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing* 17, 2 (2020), 151–178. https://doi.org/10.1007/s11633-019-1211-x

[32] Heekyung Yang and Kyungha Min. 2020. Classification of basic artistic media based on a deep convolutional approach. *The Visual Computer* 36, 3 (2020), 559–578.

[33] Lei Zhang, Meng Yang, Xiangchu Feng, Yi Ma, and D. Zhang. 2014. Collaborative Representation based Classification for Face Recognition. *International Conference on Artificial Intelligence and Software Engineering* p.21 (2014).

[34] B. Zhao, Jiashi Feng, Xiao Wu, and S. Yan. 2017. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing* 14 (2017), 119–135.