Designing an Automated Identity Verification System based on Deep Neural Networks: a Methodological Framework

Marta Donno[1], Alessandra Andreozzi[1], and Antonella Martini[1]

[1]School of Engineering
University of Pisa, Pisa, Italy
m.donno@studenti.unipi.it
alessandra.andreozzi@phd.unipi.it
antonella.martini@unipi.it

Abstract. In the current Digital Era, ensuring identity security is a priority for companies. This paper proposes a methodological framework for designing an automated online identification system by leveraging deep learning techniques.

Keywords: System Design, Methodology, Identity Verification, Deep Learning, Deep Neural Networks, Image Forensics

1 Introduction

Nowadays more and more companies offer services to their customers through online processes. This is why electronic identification[1] plays a crucial role in the user verification process. Identity Documents (ID) are used as proof of identity in many online activities, such as financial transactions. However, the phenomenon of counterfeiting IDs for personal gain is becoming increasingly common, due to powerful and easy-to-access image manipulation software, even for beginner users. In the Image Forensics domain[2], Artificial Intelligence (AI) techniques and in particular, deep learning algorithms, can be leveraged to automate the identity verification process, by independently recognizing the difference between valid and invalid digital ID samples. However, the methodical development of a deep learning-based system performing this functionality is highly complex and challenging in the real world, for several reasons. Firstly, a complete verification of identity through IDs requires the joint development of three subsystems: *(1)* Integrity Check; *(2)* Forgery Detection; *(3)* Face Detection and Matching. They represent the three main tasks of the identity verification process. Not all of these are fully addressed and solved at present by researchers in applied deep learning to Image Forensics issues [11], [12], [13]. Secondly, as stated in [14]: "(...) *practical work in deep learning on novel tasks without existing baselines remains challenging*". In this view, the motivation for this work stems in identifying a framework to address new

[1] Electronic identification is referred to a digital way to demonstrate the identity of a person or organization in order to execute online transactions.
[2] Image Forensics domain is a specific area of knowledge that aims to verify the authenticity and integrity of digital files.

problems in industrial applications that involve the development of complex deep learning models for users' identification goals.
Table 1 below summarizes the state of the art on Deep Neural Networks (DNNs) architectures and methodologies that are mostly used for dealing with the tasks in scope.

Table 1. Summary of the state of the art for Integrity Check, Forgery Detection, and Face Detection and Matching.

Task	*DNNs Models and Methodologies*
Integrity Check	There have not been released previous studies in this regard, except those that aim to assign a quality score of the image through the MobileNet [15] model.
Forgery Detection	• Hybrid encoder-decoder LSTM [18] • Two-stream Faster R-CNN [19] • Siamese Architecture [8]
Face Detection and Matching	Five-steps model: (i) Multi-Task Cascaded Neural Network; (ii) Face Alignment; (iii) Deep Convolutional Neural Network; (iv) Face embedding and normalization; (v) Faces comparison Two techniques are also proposed to improve the model's performances: • To merge the features extracted from the original images and those extracted from the mirrored ones by element-wise summation; • To use a face-similarity histogram for computing features distance.

This paper will explore the specific challenges arising in a real-world project that requires the development of an automatic identity verification system. It is based on a case study in the gambling industry, and it proposes a methodology for addressing the problem in a structured and replicable way.

2 Problem Setting

An automatic identity verification system is composed of three sub-systems: *(1)* Integrity Check; *(2)* Forgery Detection; and *(3)* Face Detection and Matching. It was required to assess the feasibility of the design and development of three deep learning-based models for each of them. All the models would be integrated into a single, final system, that can activate a warning on IDs images when they do not meet the previously required requirements. The Integrity Check task is performed to certify legibility and completeness[3] for the ID image. Forgery Detection is the process aimed to determine the authenticity of the ID image, by identifying any digital modifications. These manipulations leave inconsistencies in the image that are invisible to the human eye, but an intelligent algorithm can be trained to identify three types of tampering: *(i)* Splicing[4]; *(ii)* Copy and Move[5]; *(iii)* Removal[6]. In Face Detection and Matching the goal is to evaluate whether the person who claims to be the document's owner actually holds the ID. To perform this check, the user must upload a second image containing his/her

[3] Legibility refers to the quality of the image: it concerns the absence of blurring and movement, that are conditions that do not allow the correct reading of identification fields. As far as completeness is regarded, it mainly refers to the layout of the document, with attention to the presence of all standard fields.
[4] We talk about Splicing when a portion of an image is copied and pasted into another image, for example by taking a person's photo from a document and pasting it into another document.
[5] We have Copy and Move when a portion of an image is copied and pasted into the same image.
[6] Removal is when a portion of an image is deleted, for example by removing a second name from the ID.

selfie with the visible document in the hand. A deep learning model can compare the user's selfie and the document photo to determine facial similarity.

3 Methodological Framework

Automated identity verification can be faced up by exploiting DNNs to address the tasks of *(1)* Integrity Check; *(2)* Forgery Detection; *(3)* Face Detection and Matching. To approach and solve each of them, it has been developed an evidence-based methodology, consisting of six steps *(Six-Steps Methodology)* (see yellow boxes in Figure 1). Every step of the general framework has been applied to each task to implement a specific deep learning-based solution. In the following sections, the steps of the methodology will be described in detail, while Section 4 and Section 5 will report results and insights from practical application.

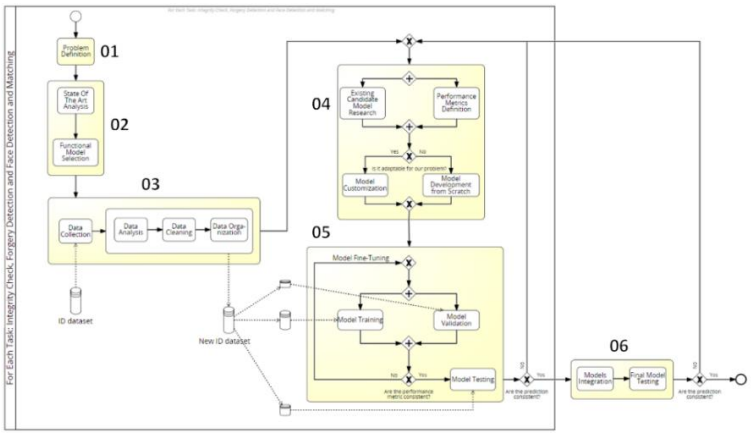

Fig. 1 Designing and developing an automated identity verification system: *Six-Steps Methodology.*

3.1 Problem Definition

The first step is to formally define the problem to be solved. The tasks executed for automated ID verification can be treated as binary classification problems, one of the most common deep learning tasks on image data. In these kind of problems, models can differentiate images between two classes. The classes depend on the task to be solved. For example, we can have valid/invalid ID image, or authentic/manipulated ID image.

3.2 Functional Model Analysis, Evaluation, and Selection

State of the art. State of the art study allows us to understand the most used and the best performing deep learning models that have been already studied and developed for the required tasks by the scientific comm, together with the relative success and failure rates.

Functional Model Selection. The model selection activity aims to determine the final architecture to be used, given a set of candidate models. In the practice, it is essential to pay attention to available resources in terms of (i) dataset, (ii) time, and (iii) hardware. These variables influence the model's output performances, in particular, the training dataset availability is crucial. The accessible training dataset should be large enough to allow the network to learn the required task, and it can be binding in the choice of the learning algorithm.

3.3 Data Processing

Data Collection. The data collection activity aims to collect data to run deep learning models. While for the Integrity Check and Face Detection and Matching tasks suitable datasets are available (MIDV 500 [1], Microsoft Celeb [2], and LFW [3]), for Forgery Detection the task complexity and lack of data led to severe limitations in completing the scheduled implementation. Determining the dataset size in terms of the number of images represents an important step in the Data Collection activity. In general, according to the amount of training data available, two macro-approaches can be distinguished: *(i)* if a large[7] dataset is available, simple algorithms and little hand-engineering[8] methods are used for data preparation, while transfer learning[9] methods are applied to obtain the desired model performances (e.g., model fine-tuning[10]) ; *(ii)* if a little dataset is available, a lot of hand-engineering is done to obtain the required model performances.

Data Preparation. The Data Preparation phase consists of three activities: *(1)* Data Analysis, *(2)* Data Cleaning, and *(3)* Data Organization. In the first one, the dataset structure is inspected to discover useful information. Data Cleaning is executed for *(i)* removing all unwanted samples, as they are irrelevant for use; *(ii)* removing duplicates, as insignificant for DNNs models. Data Organization is responsible for effectively organizing data, to use them as input for deep learning models. For each of the three required tasks for implementing the automatic verification system, the final dataset is always organized into two subsets that are valid/invalid, authentic/manipulated, and the same person/different person for Integrity Check, Forgery Detection and Face Detection and Matching respectively.

---

[7] In terms of number of images.

[8] It is the activity of manually working on the dataset organization (i.e., the way the data are fed into the network), and on the network parameters, due to the limited training data available.

[9] In transfer learning (or domain adaptation) concept, the model is pre-trained on massive dataset to learn general features for the required task. This allows to obtain better performances through the knowledge transfer from a task to a similar one where small data is available.

[10] Fine-tuning can be considered a transfer learning-based approach. In practice, an existing network already trained on a large (generally, open-source) dataset is fine-tuned by continuing the training on a smaller dataset that is not different in context to the original dataset, and changing the model learning rate to a smaller one. In this way, the already learned features from the pre-trained model become relevant to the specific classification problem.

3.4 Model Implementation

Existing Candidate Model Research. This phase is essential since many Image Forensics problems have a limited amount of available data, and training a network from scratch requires adequate computational resources and time. Following a transfer learning approach, if a neural network has high performance on a problem resolution, it often works well on similar problems. For this reason, it is better to start from an existing, already developed architecture code, by using the open-source implementation as a starting point, if available. This allows for faster training, validation, and testing phases.

Model Development. This phase aims at defining the most suitable neural network architectures, both from theoretical and code availability points of view. Each selected architecture for Integrity Check, Forgery Detection, and Face Detection and Matching tasks can be composed of one or more neural networks.

Performance Metrics Definition. Evaluation metrics are required in classification problems and they vary depending on the dataset and the model's architecture. Each of the three tasks uses different evaluation metrics [4]: *(i)* Precision, Recall and Area Under the Curve (AUC) for the Integrity Check task; *(ii)* Mean Average Precision (mAP), Receiver Operating Characteristic (ROC) Curve, Area Under the Curve (AUC) for the Forgery Detection task; *(iii)* Accuracy for the Face Detection and Matching task.

3.5 Model Training, Validation, and Testing

To perform training, validation, and testing operations the dataset is split into three parts depending on the amount of data available. The training dataset represents the sample used to train the model, and it learns from this data. The validation dataset is used to fine-tune the model hyperparameters, and this process allows us to check the model learning progress. Before starting a training procedure with DNNs, it is necessary to determine parameters: (1) Number of epochs[11]; (2) Batch size[12]; (3) Early stopping patience[13]; (4) Number of classes[14]. During training and validation, loss function[15] trends are monitored to avoid overfitting[16]. Finally, the testing dataset is used to provide an unbiased evaluation of the final model fitness on the training dataset. The testing set has never been seen by the model, and it generates the final model valuation metrics.

3.6 Models Integration and Final Testing

Lastly, the three tasks that make up the automated system must be integrated to generate a final output that ensures trustworthy user identification. For the case under analysis,

[11] How many times the algorithm sees the training set of data considering forward and backward propagation. Every time the algorithm processes all the samples in the dataset, an epoch is completed.
[12] The number of training examples in a forward/backward step of the learning process.
[13] The number of epochs to wait before early stop the validation set if there is no progress in loss performance.
[14] The number of homogeneous groups in which the deep learning model can classify the input images.
[15] In a deep neural network learning phase, the error is calculated as the difference between the actual output and the predicted output. The function that is used to compute this error is called loss function.
[16] Overfitting occurs when the loss function trend is divergent. It means that the model does not generalize well to unseen data.This is because the model adapts to features that are specific only to the training set.

integration occurs simply by recalling the models in sequence. To deploy the integrated system, the suggested guideline is to take advantage of a Docker platform[17].

4 Implementing the Six-Steps Methodology

Our six methodological steps are listed specifically for the tasks of Integrity Check, Forgery Detection and Face Detection, and Matching in Table 2, Table 3, and Table 4 respectively.

Table 2. Phases**,** Sub-phases, Key Facts, and Results for Integrity Check Task.

Phase	*Sub-phase*	*Key facts*	*Results*
Problem Definition		- Binary classification problem in supervised learning; - Check document legibility and completeness	Image classification: - 1: valid image - 0: invalid image
Functional Model Analysis, Evaluation and Selection	State of the art	- Integrity Check is not addressed in image analytic field; -A visual-based method for document classification can be adapted to the Integrity Check problem	Three-steps approach: - Local features extraction; - Local features aggregation into a global descriptor; - Global image descriptor classification
	Functional Model Selection	Selected models allow legibility and the completeness check for the uploaded image	- Quality estimator; - Segmentation; - NetVLAD [6]
Data Processing	Data Collection	Open-source dataset available	Dataset MIDV-500 [1]
	Data Preparation	- Dataset analysis; - Dataset cleaning by removing documents with different alphabets, layout or type of document; - Dataset organization into two folders: - Valid: images conform to 6 requirements (no cut, no reflections, no objects above, no blurring, every number/letter is clear, no interpretations); - Invalid: the other ones	A clean and well-organized dataset into two labeled folders: - Valid - Invalid

[17] A Docker is a complete platform which allows to contain the deep learning model.

Phase	*Sub-phase*	*Key facts*	*Results*
Model Implementation	Existing Candidate Model Research	The hybrid approach is used: transfer learning, customize existing models, and implement other models from scratch	- Quality estimator [5]: trained via transfer learning on ImageNet dataset; - Segmentation: customized on code implementation available; - NetVLAD [6] architectures: implemented from scratch
	Model Development	- Cascaded approach; - Written in Python 3.6; - Libraries used for code implementation: Numpy, TensorFlow 2.0, Tensorboard	- Quality estimator: Mobile Net [15]; - Segmentation: U-Net [16]; - NetVLAD: cropped VGG-16 [17], NetVLAD pooling layer, PCA[18], and a fully connected network [19]for classification
	Performance Metrics Definition	- High performance required to avoid False Positive[20] (FP) cases; - Monitor True Positives[21] (TP)	Metrics [4]: - Recall - Precision - AUC
Model Training, Validation, and Testing		Each architecture is trained, validated, and tested individually	- 2139 training images, 611 validation image, 200 epochs, 2 classes and 32 as batch size; - 306 test images: Precision: 95.2%, Recall: 94.5%, AUC: 97.16%
Models Integration and Final Testing		- Integration in sequence through a model recall; - Testing of the final output	Integrity check final single model

Table 3. Phases**,** Sub-phases, Key Facts and Results for Forgery Detection Task.

Phase	*Sub-phase*	*Key facts*	*Results*
Problem Definition		Object Detection problem	Object detection (classification and localization): - Detect forgeries in the image; - Localize in which pixels the manipulation occurs
Functional Model Analysis,	State of the art	- Forgery Detection is not a solved problem in the scientific community, it is in an experimental phase;	Three-steps approach: - Split the image into patches; - Compute a similarity score [8] between patches; Create a

[18] Principal Component Analysis (PCA) is a technique for performing linear dimensionality reduction. It is utilized for extracting information from a high-dimensional space by projecting it into a lower-dimensional sub-space.

[19] A fully connected network can be considered a particular type of deep neural network. In each layer, all the neurons are full connections with the neurons in the next layer. This implies a a large number of parameters, therefore it is generally expensive to train.

[20] We have a False Positive when the image is classified as "valid" by the model, but it is not actually a valid image. In the Integrity Check task, accepting an unreadable or incomplete photo as valid can have strong repercussion on the Forgery Detection task, that involves tampering recognition.

[21] We have a True Positive when the image is classified as "valid" by the model and it is actually a valid image.

Phase	*Sub-phase*	*Key facts*	*Results*
Evaluation and Selection		- Graph-base method for forgery localization	cluster-based graph.
	Functional Model Selection	The models selected allow image manipulations' checking and localizations	- Forensics similarity score [8]; - Graph with partitioned communities according to the similarity score [9]
Data Processing	Data Collection	Dataset requirements: - Images with all the manipulation cases (Splicing, Copy and Move, Removal); - Images with labeled manipulation	Dataset MIDV-500 [1] and DRESDEN Dataset [7]
	Data Preparation	- DRESDEN [7] dataset analysis; - Synthetic Dataset creation; - Dataset organization into two folders: Manipulated, Authentic	A synthetic and well-organized dataset into two labeled folders: - Manipulated; - Authentic
Model Implementation	Existing candidate Models Research	Model implementation from scratch.	- Forensic Similarity [8]; - Forensic Similarity Graph [9]
	Model Development	Cascaded approach	- Forensic Similarity [8]: MISLnet and a 3-layer similarity network; - Forensic Similarity Graph [9]: graph construction with vertex (patches) and weighted edges (similarity score)
	Performance Metric Definition	Reduce False Positive (FP) and False Negative[22] (FN)	Metrics [4]: - mAP, ROC, AUC
Model Training, Validation, and Testing		Guideline: - Training, validation, and test phase are performed in sequence	Not performed yet for three factors: - Time constraints; - Lack of data availability; - Technical constraints
Models Integration and Final Testing		Guideline: - Integration in sequence through a model recall; - Testing of the final output	Forgery Detection final single model.

Table 4. Phases**,** Sub-phases, Key Facts, and Results for Face Detection and Matching Task.

Phase	*Sub-phase*	*Key facts*	*Results*
Problem Definition		Face verification problem	Face verification: - 1: faces belong to the same person;

[22] We have a False Negative when the image is classified as "valid" by the model but it is not a valid image.

Phase	*Sub-phase*	*Key facts*	*Results*
			- 0: faces belong to a different person
Functional Model Analysis, Evaluation and Selection	State of the art	Face Detection and Matching is a fully solved problem in the face recognition field	Three-steps approach: - Face and landmark detector; - Face alignment; - Deep features extraction and face matching
	Functional Model Selection	The models selected allow us to check if the person who claims to be the owner of the document is its possessor	- Face and landmark detection; - Face embedding; - Classification
Data Processing	Data Collection	Dataset requirements: - Images with people face in different angles; - Images with people face in different facial expressions	Dataset Microsoft Celeb [2] and LFW Dataset [3]

	Data Preparation	- Microsoft Celeb [2] and LFW [3] dataset analysis;
		- Microsoft Celeb [2] and LFW [3] dataset cleaning;
		- Selfie-ID dataset creation;
		- Dataset organization:
		  - A folder for each person in the dataset;
		  - Images associated with the person in each folder
			Cleaned and well-organized dataset into *n* labeled folders, containing different images with different people expressions and angles

Model Implementation	Existing Candidate Model Research	"Ready-to-use" and available model implementation	- Face and landmark detection; - Face embedding; - Classification
	Model Development	- Cascaded approach; - Programming languages and libraries to be used for implementation: Python 3.6, Numpy, OpenCV 2.0, PyTorch	- Face and landmark detection: Multi-Task Cascaded Neural Network [10]; - Face embedding: face alignment, image flip and merge, face ResNet neural network; - Classification: cosine similarity
Model Implementation	Performance Metrics Definition	High accuracy in matching performances	Metric: - Accuracy

Phase	*Sub-phase*	*Key facts*	*Results*
Model Training, Validation, and Testing		- Each architecture is already trained and validated in the model's implementation available; - The test is performed individually	- Over 4M train and validation images from Microsoft Celeb and LFW dataset; - 100 test images: Accuracy: 86%
Models Integration and Final Testing		- Integration in sequence through a model recall; - Testing of the final output	- Final model for Face Detection and Matching task

The application of this methodology allowed us to achieve high performances for the Integrity Check task, with a Precision of 95.2%, a Recall of 94.5%, and AUC=97.16%. The developed solution is characterized by an innovative approach with three cascaded models: (i) quality estimation, which verifies image usability; (ii) segmentation, which removes document background; and (iii) NetVLAD [6], which extracts and aggregates local image descriptors into a global one. The latter feeds a fully connected network that classifies the image into valid or invalid. The design of the NetVLAD model is a ground-breaking solution because it has never been applied to an Integrity Check task. For the Forgery Detection task instead, it was not possible to fully implement the selected architectures, for the reasons explained in Table 4 and Section 5. Nevertheless, the methodology application led to a complete feasibility study on this task. Finally, for Face Detection and Matching an Accuracy of 86% was achieved, allowing us to effectively deliver this functionality.

5 Conclusions

In this work, we identified a common development methodology for approaching the ID identification problem through deep learning techniques. This methodology can be adopted as a guideline in assessing the practical feasibility of deep learning tasks in the industrial world, especially in domains where innovative methods or final datasets are not seen yet. Our deep learning implementations can potentially achieve high performances by executing repetitive, or difficult tasks in on-line identity validation. A practical business case on which we put into action the framework regarded the digitalization of the user identification process for a leading Italian company in the gambling industry. Thanks to the engagement of the *Six-Steps Methodology* for this specific case, it was possible to inductively obtain the key drivers for the project's success. In this respect, Table 5 below highlights the common critical factors for each of the three tasks and it reports a multi-criteria comparison, grouped by exogenous and endogenous factors. As noted, the availability of an appropriate open-source dataset resulted to be the key success factor for delivering the Integrity Check functionality. The dataset accessibility strongly contributed to the model performance and allowed the implementation of a solution never explored so far, filling the gap between cutting-edge research in the field and practical application. As far as Forgery Detection is concerned, the biggest challenge resulted to be the lack of a suitable dataset to train deep learning models.

Since Face Detection and Matching can be considered a solved task within the identity verification area, successful results were achieved, as reported in Table 4. By implementing the DNNs architectures explained in Tables 2, 3, and 4, respectively, we identified potential benefits for the client company and its customers (see Table 6). A powerful example of a quantitative result concerns resource savings, in terms of time[23]. Our solution has the potential to restructure the user verification process from a manual into an automated one, increasing by 44% of the ID images checked per hour. In light of these results, we are confident that the usage of deep learning-based systems is the right direction to implement an automatic verification system. Our formalized framework (*Six-Steps Methodology*) for system design could further help managers and practitioners to systematically approach a complex case, in an uncertain and constantly evolving context from the scientific and technological point of view.

Table 5. Multi-criteria comparison for Integrity Check, Forgery Detection, and Face Detection and Matching.

CRITERIA / TASK	EXOGENOUS FACTORS		ENDOGENOUS FACTORS	
	Problem addressed by the scientific community	***Problem solved by the scientific community and well documented in the literature***	***Data-set availability and its access***	***Code implementation availability***
INTEGRITY CHECK	No	No	Yes	Partially[24]
FORGERY DETECTION	Yes	No	No	No
FACE DETECTION AND MATCHING	Yes	Yes	Partially[25]	Yes

Table 6. Benefits from an Automatic Identity Verification process.

Increase productivity	Fast document checking, reducing the time spent on every ID image.
Enhance reporting creation	Automatic reports creation, accurate and reliable document structured, and precise process indicators are available.
Reduce fraud attempts	Wide type of tampering detection and error rates reduction. Fraudulent behavior prevention through suspicious pattern analysis.
Save costs	Operating and management costs reduction for faster verification made by an intelligent algorithm and saving in compliance cost.
Data-driven decision	Base the decision-making process on a measured and verifiable number as well as on high-quality information automatically aggregated.
Increase customer experience	Immediate identity validation experience for registration time reduction and error rate abatement.
Create chat-bot services	Major customer engagement with interactive interfaces.

[23] In this case, time savings means taking less time to perform all the automatic identity verifcation process' tasks. Time performance can be assessed by measuring the ID image checked per hour.
[24] The open source code available for the Integrity Check task is only relating to the estimation of image quality. The remaining part was implemented by us from scratch.
[25] The open source dataset available for the task of Face Detection and Matching is the dataset containing people faces. To perform the Face Matching we had to create a new dataset with faces of people and document in their hand.

References

1. Arlazarov, V. V., Bulatov, K. B., Chernov, T. S., & Arlazarov, V. L.: MIDV-500: a dataset for identity document analysis and recognition on mobile devices in video stream. Компьютерная оптика 43(5), (2016, October).
2. Guo, Y., Zhang, L., Hu, Y., He, X., & Gao, J.: Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In European conference on computer vision, pp. 87-102. Springer, Cham (2016, October).
3. Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments, (2008, October).
4. Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).
5. Talebi, Hossein, and Peyman Milanfar. "NIMA: Neural image assessment." IEEE Transactions on Image Processing 27.8: 3998-4011, (2018).
6. Arandjelovic, Relja, et al. "NetVLAD: CNN architecture for weakly supervised place recognition." Proceedings of the IEEE conference on computer vision and pattern recognition, (2016).
7. Gloe, Thomas, and Rainer Böhme. "The Dresden image database for benchmarking digital image forensics." Journal of Digital Forensic Practice 3.2-4: 150-159, (2010).
8. Mayer, Owen, and Matthew C. Stamm. "Forensic Similarity for Digital Images." IEEE Transactions on Information Forensics and Security 15: 1331-1346, (2019).
9. Mayer, Owen, and Matthew C. Stamm. "Exposing Fake Images with Forensic Similarity Graphs." arXiv preprint arXiv:1912.02861 (2019).
10. Zhang, Kaipeng, et al. "Joint face detection and alignment using multitask cascaded convolutional networks." IEEE Signal Processing Letters 23.10: 1499-1503, (2016).
11. Stamm, Matthew C., Min Wu, and KJ Ray Liu. "Information forensics: An overview of the first decade." IEEE access 1: 167-200, (2013).
12. Bappy, Jawadul H., et al. "Exploiting spatial structure for localizing manipulated image regions." Proceedings of the IEEE international conference on computer vision. 2017.
13. Huh, Minyoung, et al. "Fighting fake news: Image splice detection via learned self-consistency." Proceedings of the European Conference on Computer Vision (ECCV), 2018.
14. Stadelmann, T., Amirian, M., Arabaci, I., Arnold, M., Duivesteijn, F. F., Elezi, I., Geiger, M., Lörwald, S. Meier, B. B., Rombach, K., & Tuggener, L. Deep Learning in the Wild. In: Proceedings of the 8th IAPR TC 3 Workshop on Artificial Neural Networks for Pattern Recognition (ANNPR'18), Siena, Italy, September 19-21, (2018).
15. Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
16. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.
17. Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
18. Bappy, Jawadul H., et al. "Exploiting spatial structure for localizing manipulated image regions." Proceedings of the IEEE international conference on computer vision. 2017.
19. Zhu, Xinshan, et al. "A deep learning approach to patch-based image inpainting forensics." Signal Processing: Image Communication, 67: 90-99 (2018).