A Critical Look into Quantization Table Generalization Capabilities of CNN-based Double JPEG Compression Detection

Elena Rodríguez-Lois, David Vázquez-Padín, Fernando Pérez-González, Pedro Comesaña-Alfaro
Signal Theory and Communications Department
atlanTTic Research Center, University of Vigo, E.E. de Telecomunicación, 36310 Vigo, Spain
{erodriguez, dvazquez, fperez, pcomesan}@gts.uvigo.es

***Abstract*—Double JPEG compression detection has become a core issue in image forensics, as it provides information about the processing history of the image and its authenticity. Several recent works address this problem by exploiting the potential of CNNs to achieve state-of-the-art performance on test datasets. Unfortunately, those schemes are typically tailored to their specific training conditions and suffer a significant drop of performance in real-life scenarios. This paper aims at assessing the influence of quantization table mismatch (with regards to those seen in training) in the detection of double JPEG compression. Experimental results show inconsistency between different sets of quantization tables, with trained models yielding significantly worse results on unknown sets. This effect is also evident in a more realistic setting, where it appears to be more noticeable for sources falling in operating regions with greater inconsistency.**

***Index Terms*—Image forensics, Double JPEG compression, Convolutional Neural Networks, Source heterogeneity**

I. Introduction

With the current availability of digital cameras and social networking sites in which images can be widely spread, an interest in detecting image manipulations has been rapidly growing. Moreover, with the evergrowing list of camera models and processing operations (such as images being uploaded to a cloud storage service like Google Photos), this greater accessibility also results in a wider variety of sources generating digital images. Many solutions to the manipulation detection problem have been proposed that focus on the detection of double JPEG-compressed patches. Early works were built upon hand-crafted features extracted from JPEG images, including features derived from the pixels and DCT coefficients histograms [1], but these methods were often less effective, for instance, when the second JPEG compression had a lower Quality Factor (QF) than the first. Since then, the focus has shifted towards CNN-based architectures, such as [2]–[4], with promising results. However, as these new classification features are learned directly from the training data, this approach is more susceptible to overfitting to the dataset characteristics, resulting in poor generalization capabilities to unknown sources. In the field of steganalysis this problem is known as the Cover-Source Mismatch (CSM), and has been receiving increasing attention as of late with the ALASKA contest [5]. The two main strategies to deal with CSM in steganalysis are the atomistic approach (reducing its effect by using similar sources with comparable characteristics) and the holistic approach (using a diverse enough dataset so that the solution will be less dependant on the source) [6]. The latter more closely resembles the strategy used by some forensic detectors such as [4], as it requires no prior knowledge of the image source.

This work was partially funded by the European Union H2020 Framework Programme under project UNCOVER (proj. no. 101021687), and by the European Regional Development Fund (FEDER) and Xunta de Galicia under project "Grupos de Referencia Competitiva" (ED431C 2021/47).

The overall goal of this paper is to provide a deeper look into the generalization capabilities of such CNN-based double JPEG compression detectors, particularly [4] due to its similarity to the holistic approach. Generalization w.r.t. quantization tables is an imperative need in these detectors, as some sources use image-adaptive quantization tables optimized for the image content [7], making it extremely likely to encounter unknown tables in a practical setting. Experimental results that highlight the relevance of this quantization table mismatch are reported, and its impact is also evaluated in a realistic scenario using actual image-adaptive quantization tables.

The rest of the paper is organized as follows: Section II introduces prior concepts that this work relies upon, and describes the main contributions of [4]. Section III includes the description of the experimental methodology, defining the quantization table subsets considered, explaining the generation of the different image datasets and training of the models, and presenting the experimental results. Finally, Section IV concludes this paper.

II. Problem statement

A. Source mismatch

Assessing the impact of encountering unknown sources, as pointed out in [6], presents the challenge of discerning the effects of two different factors: the inconsistency w.r.t. the known source, and the unknown source's intrinsic difficulty. In other words, the unknown source might be intrinsically harder or easier to classify, which might overshadow the effect of the inconsistency itself. To address this, the authors in [6] define the source intrinsic dificulty as the score obtained when the detector has been trained only on that given source, and the

inconsistency as the score difference compared to evaluating that same source on a detector trained on a different one. Although these definitions are useful in steganalysis, forensic classifiers such as double JPEG compression detectors are typically trained by using images coming from several sources, potentially using different quantization tables, which makes such definitions less practical.

Taking this into account, an attempt is still made in this paper to estimate the intrinsic difficulty of a set of sources by training models on them that can be used as a reasonable baseline in the comparison. For the sake of simplicity, even though this does not strictly fit the original definitions in [6], the score of a known set of sources (namely, the set used in training) over a model will be referred to as *intrinsic difficulty* and the difference to the score over a similar model that was not trained on that set will be referred to as *inconsistency*.

B. Park et al.'s ConvNet

The authors in [4] propose a CNN-based double JPEG compression block detector, taking histogram features and quantization tables from the last JPEG compression as inputs. The histogram features are first extracted from the Y channel of the decompressed image, followed by a deep convolutional neural network consisting of four convolutional layers, three max pooling layers, and lastly, three fully connected layers in which the quantization table information is appended to each of their input feature vectors. The PyTorch implementation of this solution is available in a public repository[1] by the original authors, and it was used throughout this paper. This detector will be referred to as Park ConvNet for the sake of brevity.

Another contribution of [4] was to create a dataset of single and double JPEG-compressed blocks, with quantization tables randomly chosen from a set of 1120. These were gathered in the span of two years while operating a public forensic website, and include the standard quantization tables (i.e., scaled versions of the quantization tables in Annex K.1 of the JPEG standard [8], corresponding to $\mathrm{QF}_1, \mathrm{QF}_2 \in \{50, \ldots, 100\}$ according to the Independent JPEG Group, as defined in [9]). This diverse image dataset was used to train the original model.

Although the authors report an overall accuracy of 93.28% for the trained Park ConvNet on the validation blocks (unfortunately, there are no images left for testing on the dataset), a general metric is not sufficient on its own to accurately describe the performance of the detector. The difficulty of double JPEG compression detection depends on several parameters, such as the QF of the last JPEG compression or the relationship between the first (QF_1) and second (QF_2) JPEG quality factors, and a model may perform differently depending on the situation. To exemplify this, one can see in Table I the difference in True Positive Rate (TPR) and True Negative Rate (TNR) for the trained Park ConvNet on the validation blocks, where the Positive class corresponds to the double JPEG-compressed case. Both the TPR and TNR were calculated using only those blocks compressed with standard quantization tables. This restriction resulted in very few examples, specially for the double JPEG-compressed blocks, which complicates the accurate estimation of the model's performance; however, the differences in TPR, when $\mathrm{QF}_1 < \mathrm{QF}_2$ and $\mathrm{QF}_1 > \mathrm{QF}_2$, and TNR, when $\mathrm{QF} > 75$ and $\mathrm{QF} < 75$, are still significant.

[1] https://github.com/plok5308/DJPEG-torch

TABLE I: Original Park ConvNet's Performance in Different Operating Regions

Double Compression Detection		
Overall TPR(%)	TPR(%)| $\mathrm{QF}_1<\mathrm{QF}_2$	TPR(%)| $\mathrm{QF}_1>\mathrm{QF}_2$
90.59	90 (49/54)	68 (30/44)
Single Compression Detection		
Overall TNR(%)	TNR(%)| QF>75	TNR(%)| QF<75
95.97	97.6 (1289/1321)	91.3 (1114/1220)

As discussed in Sec. II-A, properly characterizing the performance of a model on a set of sources will be crucial for assessing the generalization capabilities of the detector. Given that a set can potentially present very different scores over blocks that meet specific conditions, it is essential that these operating points are taken into account, and they will have to be defined in a way that is applicable to the non-standard quantization tables considered.

C. Comparing quantization tables

The following semi-metric was first introduced in [10] and measures the 'dissimilarity' between two quantization tables $\mathbf{p}$, $\mathbf{q}$:

$$d^2(\mathbf{p},\mathbf{q}) = \sum_{k,l\in\{1,\ldots,8\}} \frac{1}{(k+l)^2}\left(\frac{q_{kl}-p_{kl}}{q_{kl}+p_{kl}}\right)^2, \qquad (1)$$

where p_{kl} and q_{kl} represent the (k,l)-th element of the quantization matrices $\mathbf{p}$ and $\mathbf{q}$, respectively. This semi-metric can be used to determine the closest standard quantization table to a given non-standard one, which can be taken as an estimate of its QF, as was done in [10]:

$$\hat{\mathrm{QF}}(\mathbf{q}) = \underset{\mathrm{QF}\in\{1,\ldots,100\}}{\arg\min}\ (d(\mathbf{q}, \mathbf{q}_{\mathrm{st}}(\mathrm{QF}))), \qquad (2)$$

where $\mathbf{q}_{\mathrm{st}}(\mathrm{QF})$ represents the standard quantization table corresponding to a given QF. Furthermore, the dissimilarity itself serves as a measure of the relationship between any two quantization tables used for the double JPEG blocks. For clarity in the representation of the results, the dissimilarity will be considered negative when the $\hat{\mathrm{QF}}$ of the first JPEG compression is strictly lower than that of the second JPEG compression.

III. Experimental Methodology and Results

This section presents the description of the quantization table subsets considered and the generated image datasets, as well as the experimental results evidencing the impact of the source mismatch.

A. Quantization tables subsets

Beyond the 1120 quantization tables available in [4], another set consisting of 1317 tables is considered, some of which are image-adaptive tables. These were collected from Photoshop, the HDR dataset [11], the Desden Image Database [12] and personal images. The following summarizes the quantization table subsets that will be referenced later on:

- Subset $\mathcal{P}$: Consisting of the 1120 tables from the original work in [4].
- Subset $\mathcal{A}$: Consisting of the additional 1317 tables ($\mathcal{P} \cap \mathcal{A} = \emptyset$).
- Subset $\mathcal{P} \cup \mathcal{A}$: Consisting of the total 2437 tables ($\mathcal{P} \cup \mathcal{A}$).
- Subset $\mathcal{G}$: Consisting of a random subset of 10 tables out of 334 from Google Photos in $\mathcal{A}$ ($\mathcal{G} \subset \mathcal{A}$).
- Subset $\mathcal{L}$: Consisting of a random subset of 10 tables out of 50 from the LG Nexus 5 camera in $\mathcal{A}$ ($\mathcal{L} \subset \mathcal{A}$).
- Subset $\mathcal{S}$: Consisting of a random subset of 10 tables out of 40 from the Samsung Galaxy Note 8 (SM-N950U) camera in $\mathcal{A}$ ($\mathcal{S} \subset \mathcal{A}$).
- Subset $\mathcal{K}$: Consisting of a random subset of 10 tables out of 51 from Kodak Easyshare M1063 in $\mathcal{A}$ ($\mathcal{K} \subset \mathcal{A}$)
- Subset $\mathcal{P}' \cup \mathcal{L} \cup \mathcal{G}$: Consisting of a random subset of 1100 tables from $\mathcal{P}$, and all tables from $\mathcal{L}$ and $\mathcal{G}$.
- Subset $\mathcal{P}' \cup \mathcal{S} \cup \mathcal{G}$: Consisting of a random subset of 1100 tables from $\mathcal{P}$, and all tables from $\mathcal{S}$ and $\mathcal{G}$.
- Subset $\mathcal{P}' \cup \mathcal{K} \cup \mathcal{G}$: Consisting of a random subset of 1100 tables from $\mathcal{P}$, and all tables from $\mathcal{K}$ and $\mathcal{G}$.

Note that for subsets $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$, where $\mathcal{X} \in \{\mathcal{L}, \mathcal{S}, \mathcal{K}\}$, the change w.r.t. $\mathcal{P}$ is $<2\%$ (so as to restrict the impact on the overall cost function during training), the size of the subsets is the same as $\mathcal{P}$, and all standard quantization tables were included in the subset of 1100 tables from $\mathcal{P}$.

As was done in [4], all quantization tables across all subsets quantize the luminance channel only, using an 8×8 matrix of ones for the quantization of the chrominance.

B. Image datasets generation

A number of random image datasets were constructed in order to train the models, emulating the original generation method in [4], but using in our case the RAISE8K dataset [13], as the original dataset from [4] did not inlude the uncompressed version of the blocks. Additionally, three other datasets were also constructed following different pipelines that are more common in practice. The datasets were generated as follows:

1) Random datasets: Firstly, 521,728 blocks of size 256×256 were randomly cropped from the RAW images in RAISE8K. Each block was then compressed with a randomly chosen quantization table from the subset, resulting in the single JPEG-compressed blocks, and finally was further compressed with another quantization table, selected at random again, for the double JPEG-compressed blocks. From the total of 1,043,456 blocks, 733,680 were used for training, 48,912 for validation, and 260,864 for testing (similar ratios were used in [10], providing a high number of test images to better characterize the performance of the detector). Blocks belonging to the same RAW image were always included in the same category across all datasets. Random datasets were created for the $\mathcal{P}$, $\mathcal{A}$, $\mathcal{P} \cup \mathcal{A}$ and $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$ subsets that were described previously and, for notational simplicity, will be named after them. Additionally, another dataset was needed for the $\mathcal{P} \cup \mathcal{A}$ subset with double the blocks, as will be seen in Sec. III-C, that will be referred to as $(\mathcal{P} \cup \mathcal{A})^\star$.

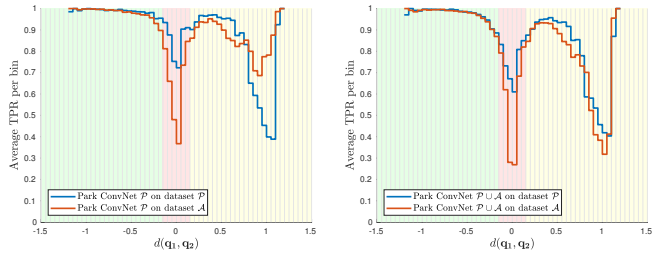
2) Pipeline datasets: In order to evaluate the impact of the quantization table inconsistency in a more realistic setting, three pipeline datasets (that will be referred to as $\mathcal{L} \rightarrow \mathcal{G}$, $\mathcal{S} \rightarrow \mathcal{G}$ and $\mathcal{K} \rightarrow \mathcal{G}$) were generated using quantization tables that were originally image-adaptive (i.e., from subsets $\mathcal{L}$, $\mathcal{S}$, and $\mathcal{K}$) and Google Photos (i.e., from subset $\mathcal{G}$), which also employs image-adaptive tables, emulating the scenario in which users upload their single JPEG-compressed images to the cloud. Using only the RAW images for the test blocks in the random datasets, 16,304 blocks of size 256×256 were randomly cropped and compressed with a randomly chosen quantization table from the given adaptive camera dataset, resulting in single JPEG-compressed blocks, and further compressed with a quantization table randomly chosen from $\mathcal{G}$ to create the double JPEG-compressed blocks.

C. Experimental results

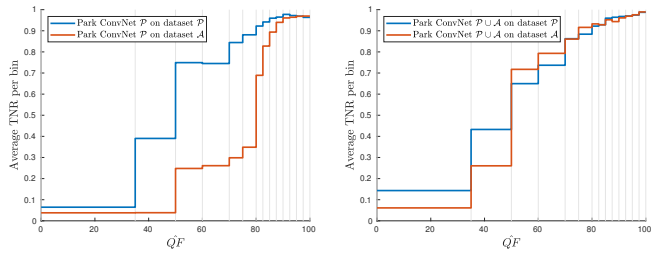
As presented in Sec. II, training a single Park ConvNet would not be enough to assess the impact of encountering unknown sources, as some baseline for the intrinsic difficulty of such sources is needed. Because of this, a different Park ConvNet was trained for each of the generated random image datasets ($\mathcal{P}$, $\mathcal{A}$, $\mathcal{P} \cup \mathcal{A}$, $(\mathcal{P} \cup \mathcal{A})^\star$, $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$), and for the sake of clarity, each will be named according to the dataset it was trained on. The different models were trained using the original configuration of the code in the public repository, with a batch size of 32 for 10 epochs, after which the weights yielding the highest overall accuracy in the validation dataset were saved.

1) Inconsistency analysis: Taking subsets $\mathcal{P}$ and $\mathcal{A}$, the aim of this analysis is to assess whether the inconsistency they would present w.r.t. each other is significant in a Park ConvNet model. In order to more accurately represent the detector's response to a given dataset, the dissimilarity semi-metric and the estimated QFs of the quantization tables are used to define the following operating regions:

- In double JPEG block detection: 60 uniform bins are considered from $d(\mathbf{q_1}, \mathbf{q_2}) = -1.5$ to $d(\mathbf{q_1}, \mathbf{q_2}) = 1.5$, with $\mathbf{q_1}$, $\mathbf{q_2}$, the quantization tables used in the first and second JPEG compressions. In more general terms, three intervals can be defined as $d(\mathbf{q_1}, \mathbf{q_2}) < -\delta$, $d(\mathbf{q_1}, \mathbf{q_2}) \in [-\delta, \delta]$, $d(\mathbf{q_1}, \mathbf{q_2}) > \delta$ (and have been shaded in Figs. 1(a)-(b) and 2(a)-(b) in green, red and yellow, respectively). As double JPEG compression detection tends to be more challenging when $\mathbf{q_1}$ and $\mathbf{q_2}$ are very similar, δ can be visually determined according to the score drop around $d(\mathbf{q_1}, \mathbf{q_2}) \sim 0$, and throughout this paper it was taken as $\delta = 0.15$.

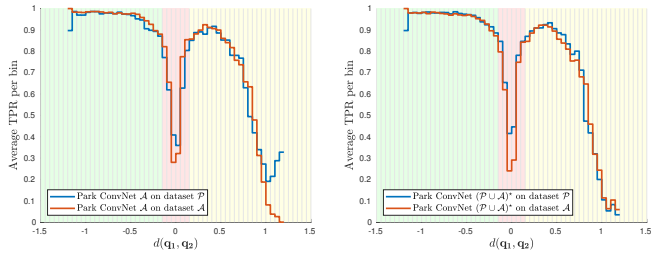


(a) Park ConvNet $\mathcal{P}$: TPR (b) Park ConvNet $\mathcal{P} \cup \mathcal{A}$: TPR

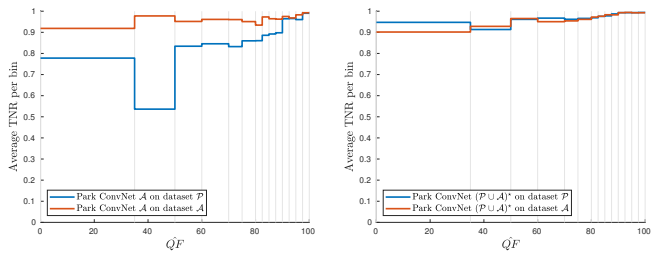


(c) Park ConvNet $\mathcal{P}$: TNR (d) Park ConvNet $\mathcal{P} \cup \mathcal{A}$: TNR

Fig. 1: Experimental TPR and TNR for Park ConvNet $\mathcal{P}$ and $\mathcal{P} \cup \mathcal{A}$, evaluated on test datasets $\mathcal{P}$ and $\mathcal{A}$



(a) Park ConvNet $\mathcal{A}$: TPR (b) Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$: TPR



(c) Park ConvNet $\mathcal{A}$: TNR (d) Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$: TNR

Fig. 2: Experimental TPR and TNR for Park ConvNet $\mathcal{A}$ and $(\mathcal{P} \cup \mathcal{A})^\star$, evaluated on test datasets $\mathcal{P}$ and $\mathcal{A}$

- In single JPEG block detection: 14 non-uniform bins are used, according to edges $\hat{\text{QF}}(\mathbf{q}) = \{0,\ 35,\ 50,\ 60,\ 70,\ 75,\ 80,\ 82.5,\ 85,\ 87.5,\ 90,\ 92.5,\ 95,\ 97.5,\ 100\}$, with $\mathbf{q}$ being the quantization table used in the single JPEG compression.

For each of the operating regions of the detector, either the TPR or TNR was calculated according to the test blocks that fit the criteria.

One of the challenges of this comparison was the contrasting strategies the different Park ConvNets were learning depending on the training dataset, which made their performance fundamentally different across the operating regions. Since the model learns by optimizing its weights according to a cost function over a given dataset, there might be different local minima that yield similar overall results, which only stresses the importance of considering the different operating conditions while evaluating the performance of a detector. In order to preserve the original Park ConvNet nothing was changed in the training of the models, and they were free to converge to whatever strategy resulted. This can be seen for instance when comparing Figs. 1(c) and 2(c), where very different TNRs can be observed across the $\hat{\text{QF}}$ while the overall accuracy of those models on the test datasets was very close: 90.14% (Park ConvNet $\mathcal{P}$ tested on dataset $\mathcal{P}$) and 90.03% (Park ConvNet $\mathcal{A}$ tested on dataset $\mathcal{A}$). Because of the potential different strategies that can be learned, only models with similar scores across the operating regions on the known datasets where compared to estimate the inconsistency.

Figure 1 shows the performance of Park ConvNets $\mathcal{P}$ and $\mathcal{P} \cup \mathcal{A}$, which converged to models with similar performance, sacrificing the detection of single JPEG blocks with a stronger compression ($\hat{\text{QF}} < 50$) in favour of the double JPEG blocks where the second compression is significantly stronger than the first ($d(\mathbf{q_1}, \mathbf{q_2}) > 1$). Taking Park ConvNet $\mathcal{P} \cup \mathcal{A}$ as a reference of how dataset $\mathcal{A}$ could potentially be detected (an insight of its intrinsic difficulty on this strategy), it is clear that there exists inconsistency between the two sets, specially in Fig. 1(c) w.r.t. Fig. 1(d), in the drop of single JPEG detection for $35 < \hat{\text{QF}} < 85$ on dataset $\mathcal{A}$. The TPR, however, increases for $d(\mathbf{q_1}, \mathbf{q_2}) \gg 0$, most likely as a result of the decreased TNR for single JPEG blocks with stronger compressions. The overall accuracy of dataset $\mathcal{A}$ falls from 88.54% (Park ConvNet $\mathcal{P} \cup \mathcal{A}$) to 82.91% (Park ConvNet $\mathcal{P}$).

Given that the diversity of dataset $\mathcal{P} \cup \mathcal{A}$ is higher, training on the same number of images slightly hinders the performance of Park ConvNet $\mathcal{P} \cup \mathcal{A}$, achieving a lower overall accuracy on dataset $\mathcal{P}$ (88.92%) compared to Park ConvNet $\mathcal{P}$ (90.14%). This effect can also be observed in Figs. 1(a), 1(b) in $d(\mathbf{q_1}, \mathbf{q_2}) \in [-\delta, \delta]$. Because of this, and although the exact impact of the diversity is not clear, Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$ is then trained on double the data.

Coincidentally, Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$ converged to a model whose performance resembles that of Park ConvNet $\mathcal{A}$, as can be seen in Fig. 2. This solution seems to prioritize the single JPEG detection, even when the last compression is strong, over double JPEG detection when $d(\mathbf{q_1}, \mathbf{q_2}) > 1$. As was done before, Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$ could be now taken as a reference for the intrinsic difficulty of $\mathcal{P}$ in this strategy. The inconsistency between the set of sources is again noticeable, specially in Figs. 2(c)-(d) in the drop of TNR on $\mathcal{P}$ for $\hat{\text{QF}} < 90$, but also the drop of TPR for $d(\mathbf{q_1}, \mathbf{q_2}) \in [-\delta, \delta]$ in Figs. 2(a)-(b). On the other hand, as in the previous comparison, the TPR increases for $d(\mathbf{q_1}, \mathbf{q_2}) > 1$ on $\mathcal{P}$, again most likely as a result of the TNR drop. In this case, the overall accuracy on dataset $\mathcal{P}$ falls from 91.11% (Park ConvNet $(\mathcal{P} \cup \mathcal{A})^\star$) to 85.52% (Park ConvNet $\mathcal{A}$).

TABLE II: Impact over the Pipeline Datasets

	Overall Accuracy(%) on Pipeline Datasets	
	Park ConvNet $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$	Park ConvNet $\mathcal{P}$
$\mathcal{L} \rightarrow \mathcal{G}$ $(\mathcal{X} = \mathcal{L})$	93.56	91.16
$\mathcal{S} \rightarrow \mathcal{G}$ $(\mathcal{X} = \mathcal{S})$	93.24	90.82
$\mathcal{K} \rightarrow \mathcal{G}$ $(\mathcal{X} = \mathcal{K})$	87.79	76.51

Unlike before, the overall accuracy on the known dataset (now $\mathcal{A}$) is similar for both models, with 90.67% (Park ConvNet $(\mathcal{P} \cup \mathcal{A})^{\star}$) and 90.03% (Park ConvNet $\mathcal{A}$), which highlights the importance of training over a large amount of data when taking a holistic approach.

Considering both comparisons, there seems to be a tendency for generalization to be less of a challenge in easier operating regions, such as $d(\mathbf{q_1}, \mathbf{q_2}) < -\delta$ for double JPEG compression, and $\hat{\text{QF}} > 90$ for the single JPEG-compressed case. On the contrary, single JPEG compressions where $\hat{\text{QF}} < 80$ seem to be the most affected, followed by the gap around $d(\mathbf{q_1}, \mathbf{q_2}) \in [-\delta, \delta]$ in the second strategy, as seen in Fig. 2.

2) Impact over a realistic pipeline: The presence of inconsistency is evident when training models in significantly different subsets of quantization tables, but its impact was also assessed for a more realistic scenario where the unknown quantization tables belong only to a specific test pipeline. For this purpose, Park ConvNets $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$ were trained on datasets very similar to $\mathcal{P}$ so that the overall cost function during training was not affected drastically by the new tables, as was explained in Sec. III-A.

Each pipeline dataset ($\mathcal{L} \rightarrow \mathcal{G}$, $\mathcal{S} \rightarrow \mathcal{G}$, $\mathcal{K} \rightarrow \mathcal{G}$) was evaluated on its corresponding Park ConvNet $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$ as a baseline for its intrinsic difficulty, and also Park ConvNet $\mathcal{P}$, to estimate the impact of the inconsistency. These results can be seen in Table II. On the one hand, datasets $\mathcal{L} \rightarrow \mathcal{G}$ and $\mathcal{S} \rightarrow \mathcal{G}$ present a slight drop of performance of around 2.4% when evaluated in Park ConvNet $\mathcal{P}$, but it is still significant when taking into account the small percentage that these tables account for in $\mathcal{P}' \cup \mathcal{X} \cup \mathcal{G}$. On the other hand, $\mathcal{K} \rightarrow \mathcal{G}$ suffers a greater accuracy fall of 11.28%. It is worth noting that both $\mathcal{L}$ and $\mathcal{S}$ correspond to smartphone cameras whose quantization table $\hat{\text{QF}}$s fall in the range of 82 to 85, whereas $\mathcal{K}$ belongs to a compact camera included in the Dresden Image Database [12], with a wider range of $\hat{\text{QF}}$s, some of which are as low as 57. Considering Fig. 1(c), where generalization seems to be more challenging for lower $\hat{\text{QF}}$s, subset $\mathcal{K}$ would understandably be more affected by the inconsistency.

IV. Conclusion

Current CNN-based solutions for the double JPEG compression detection are susceptible to overfitting to the dataset characteristics, as the detection features are learned directly from the data. With some sources using image-adaptive quantization tables, good generalization capabilities are key for detectors to be used in real-world images, as encountering unknown tables is very likely. Experimental results have been presented to illustrate the inconsistency between different sets of sources, evidencing that the detector does overfit to the quantization table subset used in training. Furthermore, the impact this can have on a more realistic scenario can be significant, specially in those operating regions where generalization is more challenging, as was seen in the $\mathcal{K} \rightarrow \mathcal{G}$ pipeline. Additionally, it is noted that training with a more diverse dataset also carries its own drawbacks, as a larger image dataset might be needed to achieve comparable results.

Considering all this, the problem of source mismatch should be taken into consideration in practical double JPEG compression detection, and further research is still needed to develop alternative approaches to the holistic strategy, ideally taking into account the more problematic operating regions rather than blindly increasing the diversity of the training dataset.

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 01 2013.

[2] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, 10 2016.

[3] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, "Aligned and non-aligned double JPEG detection using convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 49, p. 153–163, Nov 2017. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2017.09.003

[4] J.-S. Park, D. Cho, W. Ahn, and H.-K. Lee, "Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network," in *ECCV*, 2018.

[5] R. Cogranne, Q. Giboulot, and P. Bas, "The ALASKA steganalysis challenge: A first step towards steganalysis," in *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 125–137. [Online]. Available: https://doi.org/10.1145/3335203.3335726

[6] Q. Giboulot, R. Cogranne, D. Borghys, and P. Bas, "Effects and solutions of cover-source mismatch in image steganalysis," *Signal Process. Image Commun.*, vol. 86, p. 115888, 2020.

[7] H. T. Fung and K. J. Parker, "Design of image-adaptive quantization tables for JPEG," *Journal of Electronic Imaging*, vol. 4, no. 2, pp. 144 – 150, 1995. [Online]. Available: https://doi.org/10.1117/12.199459

[8] "Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines," CCITT, Standard, Sep. 1992.

[9] J. D. Kornblum, "Using JPEG quantization tables to identify imagery processed by software," *Digital Investigation*, vol. 5, pp. S21–S25, 2008, the Proceedings of the Eighth Annual DFRWS Conference. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1742287608000285

[10] Y. Yousfi and J. Fridrich, "JPEG steganalysis detectors scalable with respect to compression quality," *Electronic Imaging*, vol. 2020, pp. 75–1, 01 2020.

[11] O. A. Shaya, P. Yang, R. Ni, Y. Zhao, and A. Piva, "A new dataset for source identification of high dynamic range images," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3801

[12] T. Gloe and R. Böhme, "The 'Dresden Image Database' for benchmarking digital image forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 1584–1590. [Online]. Available: https://doi.org/10.1145/1774088.1774427

[13] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 219–224. [Online]. Available: https://doi.org/10.1145/2713168.2713194