

Controlled Evaluation for Numeric-Field Forgery Detection in SROIE Receipts using Fine-Tuned EfficientNetV2-S

Tousi Adak¹, Priyanshu Mukherjee¹, Sudipta Das²,
Jahir Uddin Paik³

¹Assistant Professor, Computer Science and Engineering, Pailan College of Management & Technology
^{2,3,4}B.Tech 4th Year, Computer Science and Engineering, Pailan College of Management & Technology

Abstract

Altered amounts, dates, and identifiers on retail receipts represent a practical source of expense and tax fraud. However, public receipt datasets such as Scanned Receipts OCR and Information Extraction (SROIE) contain authentic receipts but do not provide forged counterparts, limiting controlled evaluation of receipt-forgery detection methods. Chakraborty (2025) reported 97.1% accuracy using a convolutional neural network (CNN)-based receipt-forgery detection framework. Building on this research context, the present study investigates forgery detection at the numeric-field level, focusing on receipt components that directly carry financial information. Using SROIE, a receipt-grouped benchmark of paired authentic and synthetically forged numeric-field crops is constructed. Two controlled forgery mechanisms are considered: splicing a donor numeric field from the same receipt and erasing and re-rendering a field with a specified proportion of its digits changed. Authentic and forged crops undergo the same Joint Photographic Experts Group (JPEG) compression procedure, and receipt-level grouping is used for training, validation, and test partitioning at approximately 70/15/15. An ImageNet-pretrained EfficientNetV2-S is fine-tuned end-to-end using linear warm-up, cosine learning-rate decay, and exponential moving average (EMA) weights, with model selection based exclusively on validation accuracy. On the held-out test set of 4,602 crops, comprising 2,301 authentic and 2,301 forged samples, the reference detector achieves 98.09% accuracy, 99.42% precision, 96.74% recall, an F1-score of 0.9806, and an area under the receiver operating characteristic (ROC) curve (AUC) of 0.9971 at a fixed decision threshold of 0.5 in a single run with seed 42. Most remaining errors correspond to missed forgeries. The reported accuracy is numerically higher than the 97.1% reported by Chakraborty (2025), but direct comparison is limited by differences in datasets, sample construction, and evaluation protocols. Gradient-weighted Class Activation Mapping (Grad-CAM) analysis indicates attention to numeric text regions but also, in some cases, to crop boundaries and background regions, suggesting that synthetic generation artifacts may contribute to the observed performance. As the study relies on synthetic forgeries without multiple baseline architectures, ablation experiments, multiple random seeds, or real-world forged-receipt evaluation, the findings are limited to the defined synthetic tampering processes rather than general real-world forgery-detection performance.

Keywords: Receipt Forgery Detection, Document Tampering, Numeric-Field Forgery, EfficientNetV2-

S, SROIE, Grad-CAM, Synthetic Data, Receipt Image Analysis

1. Introduction

Receipts serve as proof of purchase in expense reimbursement, tax filing and auditing, and a receipt whose amount, date or item has been altered exposes all parties to financial risk [1]. Manual review does not scale, and optical character recognition (OCR)-and-match pipelines can be brittle when receipts vary in layout and print quality [1]. Image-based classifiers offer a complementary route because tampering of printed digits can leave local texture, stroke and background inconsistencies.

Within a receipt, numeric fields are the most consequential targets for manipulation. Monetary amounts such as totals, line prices and amounts paid fix the sum claimed in a reimbursement or deduction; dates determine whether an expense falls inside a claim period or tax year; and tax-related figures and transaction or invoice identifiers are used to validate a document and to match it against other records. Changing a few digits in any of these fields can alter the financial meaning of a receipt while the merchant name, layout and the rest of the page remain untouched. Such tampering is local and can be carried out with little effort, yet its impact can be large, which makes the numeric field, rather than the whole receipt, the natural unit of inspection. Field-level analysis also localizes the evidence: a decision about a single amount or date can be inspected and compared with its surroundings, whereas a receipt-level score cannot indicate which value was altered.

A central practical obstacle is the lack of labeled forgeries. The Scanned Receipts OCR and Information Extraction (SROIE) collection [6] provides scanned receipts with text-box annotations but no forged samples, although its annotations make it possible to locate numeric fields precisely. Dedicated forged-receipt data are available [2], and this work does not replace them. To the best of the authors' knowledge, they are not organized as paired authentic and forged field crops produced by a known, parameterized tampering process, which is what is needed to attribute detection to tampering evidence rather than to differences in source, compression or crop geometry; evaluation on them is left to future work (Section 11). This work addresses a complementary question in a controlled setting where the tampering is generated, exactly labeled and confined to a single field. Numeric text fields, meaning annotated text boxes that contain digits, such as amounts, dates and identifiers, are cropped from genuine receipts, and a matching forged version of each crop is generated by splicing in a donor field or by erasing the field and re-rendering it with altered digits. The field type is not modeled; the classifier sees only the cropped image. This controlled setting has a defined purpose and scope. Because the tampering process is known, the labels are exact and both classes share the same compression, two questions can be asked that a benchmark of real forgeries cannot answer cleanly. First, how reliably does an ImageNet-pretrained convolutional network separate authentic from forged numeric fields once source, compression and crop geometry are matched? Second, what evidence does it use: the digits themselves, or the crop boundaries and rendering traces that a generator can leave behind, a failure mode that the shortcut-learning literature identifies as common [26]? The results measure detection of the synthetic process and are not evidence of performance on real forgeries, and it is not claimed that real forgers produce tampering of this form.

Prior work either targets receipts through OCR, semantic or handcrafted-feature pipelines evaluated partly on private data [1, 2, 3], or localizes tampered text on whole document images with dedicated architectures and large synthetic corpora [23, 24]. The present study uses a standard backbone, and what distinguishes it is the experimental design: paired field-level samples, shared compression, receipt-grouped splits and

an inspection of the cues the model uses. Section 2 positions the work against these studies and states the resulting research gap.

The contributions of this work are:

- 1) a receipt-grouped, paired benchmark built entirely from public SROIE data, in which each forged numeric-field crop has an authentic counterpart from the same location, both classes share one Joint Photographic Experts Group (JPEG) compression process, and the forgery parameters are fully specified for reproduction;
- 2) a reference detector and evaluation protocol: an ImageNet-pretrained EfficientNetV2-S [7] with validation-only model selection and a fixed test threshold, whose architecture and training components are standard and are not claimed as novel;
- 3) an analysis of what the detector relies on, using Gradient-weighted Class Activation Mapping (Grad-CAM) [15] and an error breakdown, together with an explicit statement of what the synthetic setting can and cannot support and of the baselines, ablations and cross-generator tests needed next (Section 11).

The remainder of the paper is organized as follows: Sections 2–5 describe related work, data, method and formulation; Sections 6–8 the experimental setup, metrics and results; Sections 9–12 discussion, error analysis, limitations and conclusions.

2. Related Work and Research Gap

Document and receipt forgery detection has received increasing attention because alterations to financial documents can affect expense reimbursement, auditing, taxation, and other financial processes. Existing research has explored dedicated receipt-forgery datasets, document-level semantic analysis, deep learning-based document manipulation, and general document forgery detection. However, important challenges remain regarding the availability of realistic forged data, reproducible experimental protocols, and the generalization of deep learning models to unseen tampering patterns. Receipt-specific studies, document-tampering detectors, general image forensics, and work on synthetic data and shortcut learning are reviewed, summarized in Table 1, and followed by a statement of the research gap.

Tornés, et al. (2023) [2] introduced a dedicated receipt dataset for document forgery detection, providing an important resource for developing and evaluating automated receipt-forgery detection methods. The availability of forged receipt samples addresses a major limitation of datasets such as SROIE, which primarily contains genuine scanned receipts together with text and bounding-box annotations. In a related study, Tornés, et al. (2023) [3] investigated forged receipt detection using domain-specific ontology-based entities and relations. Their approach emphasizes the semantic structure of receipt information and the relationships between entities, demonstrating that receipt forgery can be analyzed beyond purely visual characteristics. These studies establish the importance of both suitable datasets and receipt-specific information for reliable forgery detection.

Huang, et al. (2019) [6] developed the International Conference on Document Analysis and Recognition (ICDAR) 2019 competition dataset and benchmark for scanned receipt OCR and information extraction. Although the dataset was designed primarily for OCR and information extraction rather than forgery detection, it provides structured receipt images and text-box annotations that can support downstream document-analysis applications. In the present study, the SROIE data associated with this benchmark are used as the source of genuine receipt images and numeric text-box annotations. Since the original dataset

does not provide forged versions of these fields, synthetic tampering is introduced to construct paired authentic and forged samples.

Deep learning has also been investigated for document-image manipulation and forgery. Zhao, et al. (2021) [4] studied deep learning-based forgery attacks on document images, highlighting the increasing capability of learning-based techniques to manipulate document content. Such developments further motivate the need for robust automated methods capable of distinguishing authentic document regions from manipulated ones. Sukhija, et al. (2025) [5] provided a comprehensive review of document forgery detection and summarized the development of conventional and deep learning-based approaches. The review indicates the growing role of deep learning in document-forgery analysis while also highlighting challenges associated with datasets, forgery diversity, and reliable evaluation.

A recent study by Chakraborty (2025) [1] proposed a receipt-forgery detection framework combining region detection using Detectron2, foreground extraction using GrabCut, texture descriptors such as the gray-level co-occurrence matrix (GLCM) and local binary patterns (LBP), and a convolutional neural network (CNN) whose hyperparameters were optimized using an updated Sand Cat Swarm Optimization (USCSO) strategy. The study reported 97.1% accuracy, 97.3% precision, and 97.5% recall on a test set containing 202 samples obtained from SROIE and a private dataset [1]. This work demonstrates that combining region detection, handcrafted texture information, deep learning, and metaheuristic optimization can provide an effective receipt-forgery detection pipeline. However, its evaluation involves a relatively small test set and a private dataset that is not publicly available, which limits direct reproducibility and independent comparison. Some reported performance values also differ across the manuscript, which complicates exact comparison.

Tampering with document images has also been studied as a general image-forensics problem. Early work on image manipulation detection used convolutional networks designed to pick up low-level forensic traces of splicing and copy-move operations [18, 19]; later networks such as RGB-N and ManTra-Net combined red-green-blue (RGB) evidence with noise or anomaly features to detect and localize manipulations [20, 21]; learned camera-fingerprint and self-consistency cues have been used to expose splices as well [28, 29]; and a recent overview summarizes this literature and its dependence on compression and post-processing [22]. These methods target natural photographs, where tampering leaves statistical traces in textured regions. Printed receipt digits are small, high-contrast and nearly binary, so such traces are weaker, and Qu, et al. (2023) [23] note that methods built for natural images may not work well on document tampering.

Document-specific tampering detectors address this gap directly. Qu, et al. (2023) [23] introduced DocTamper, a large synthetic collection of tampered document images, together with the Document Tampering Detector (DTD), which adds a frequency-perception head to expose compression-block inconsistencies and a curriculum-learning schedule that improves robustness to JPEG compression; they also evaluate on T-SROIE. Luo, et al. (2023) [24] argue that synthetic tampering inadequately reproduces real manipulation attributes and introduce the Real Text Manipulation dataset, on which existing detectors perform poorly. These works localize tampered text at pixel or region level on full document images and emphasize frequency-domain cues and training strategies. They do not isolate the numeric field as the unit of analysis, and they do not examine whether a classifier trained on generated tampering is using the generator's own artifacts rather than evidence of tampering.

Two further bodies of work bear on the experimental design. First, generating training data by rendering or splicing text is a long-established practice for text recognition and localization [25], and DocTamper

was likewise built by synthesizing tampering [23]; its known risk is that a model can learn the generator rather than the task. Second, the shortcut-learning literature shows that deep networks readily exploit unintended, dataset-specific cues, so that high benchmark accuracy need not reflect the intended capability [26]. In forgery detection this risk is concrete: differing JPEG histories, font rendering or crop geometry between authentic and forged samples can separate the classes without any tampering evidence. Controlling such cues, and testing whether a detector is relying on them, is therefore part of a credible evaluation protocol, and it is a point that the receipt-forgery studies in the present reference set [1, 2, 3] do not report in detail.

Table 1: Systematic Comparison with Related Work

Study	Dataset	Task	Method	Unit	Real/Synth etic	Main limitation
Chakrab orty (2025) [1]	SROIE+ private dataset	Receipt forgery detection	Detectron2, GrabCut, GLCM/LBP, CNN with USCSO	Receipt regions	Real / mixed-source	Private data and small test set (202 samples); limits reproducibility and direct comparison.
Tornés, et al. (2023) [2]	Dedicated forged-receipt dataset	Receipt forgery detection	Dedicated receipt-forgery dataset	Receipts	Real forged receipts	Not organized as the paired, parameterized numeric-field benchmark used in this study; evaluation here is future work.
Tornés, et al. (2023) [3]	Receipt-forgery data	Forged receipt detection	Domain-specific ontology-based entities and relations	Receipt semantic s	Real forged receipts	Emphasizes semantic/content-level cues rather than image-only numeric-field evidence.
Zhao, et al. (2021) [4]	Documen t images	Document forgery attack	Deep-learning-based forgery attack	Docume nt images	Synthetic	Focuses on generating/manipu lating document images rather than detecting the controlled numeric-field forgeries studied here.

Sukhija, et al. (2025) [5]	Document-forgery literature	Review	Comprehensive review of document-forgery detection	Not applicable	N/A	Provides background and synthesis rather than a new detection benchmark or experimental detector.
Qu, et al. (2023) [23]	DocTamp corpus; T-SROIE	Tampered-text detection/localization	DTD with frequency-perception head and curriculum learning	Document images / localization	Synthetic	Targets whole-document text localization; does not isolate paired numeric fields as the classification unit.
Luo, et al. (2023) [24]	Real Text Manipulation (RTM) dataset	Real text manipulation detection	Fusion baseline on real text manipulation data	Text images	Real	Addresses real manipulation, whereas the present study uses controlled synthetic tampering and field-level pairing.
This study	SROIE	Numeric-field forgery classification	ImageNet-pretrained EfficientNet V2-S reference detector	Paired numeric-field crops	Synthetic / controlled	Single reference architecture, one seed, no ablation/baseline study, and no real forged-receipt evaluation.

2.1 Research Gap

The existing literature reveals several gaps that motivate the present study. First, although dedicated forged-receipt resources have been introduced [2], publicly available receipt datasets such as SROIE do not inherently provide labeled forged samples for supervised learning. Consequently, a reproducible method is required for constructing authentic and forged samples from a publicly accessible dataset while maintaining a controlled experimental protocol.

Second, receipt-specific studies [1, 2, 3] rely on OCR, ontology-based semantics, handcrafted texture descriptors, region extraction and metaheuristic tuning, while document-tampering detectors [23, 24] operate on whole images with dedicated architectures. Neither line of work isolates the numeric field, the unit that carries the financial meaning of a receipt, as a controlled, paired classification problem in which every forged crop has an authentic counterpart cut from the same location, shares its JPEG process, and is

split by receipt. Without that pairing, differences in source, compression or crop geometry can be mistaken for tampering evidence [26].

Third, reproducibility remains an important issue. The reference study by Chakraborty (2025) [1] uses data from both SROIE and a private dataset, whereas the present work constructs the experimental data from publicly accessible SROIE receipts. In addition, the present study performs receipt-level splitting, ensuring that crops originating from the same receipt do not appear across training, validation, and test sets. This design reduces the possibility of information leakage between subsets and provides a more clearly defined evaluation protocol.

Fourth, the evaluation of such detectors seldom asks what the model has learned. Reported accuracy on generated tampering can reflect traces of the generator rather than of manipulation, and this is not examined in the present reference set [1, 2, 3]. Every forged crop is therefore paired with its authentic source, the same compression is applied to both classes, and the evidence used by the network is inspected with Grad-CAM [15], a qualitative tool whose faithfulness is itself debated [30], and with an error breakdown by outcome. An ImageNet-pretrained EfficientNetV2-S [7] is used only as a standard, well-documented backbone, because transfer from ImageNet-pretrained networks is a common and strong starting point for new image tasks [27]. The choice of backbone is not claimed as a contribution.

Accordingly, the contribution of this work is not a new network architecture or training recipe. It is a controlled, receipt-grouped, paired benchmark and evaluation protocol for numeric-field tampering built entirely from the public SROIE data, together with a qualitative inspection (Grad-CAM and an error breakdown) of the evidence the model relies on, including possible generator artifacts. The EfficientNetV2-S model serves as the reference detector within that protocol. The study does not claim superiority over Chakraborty (2025) [1], Qu, et al. (2023) [23] or Luo, et al. (2023) [24], because datasets, forgery-generation procedures, test-set sizes and protocols differ. The comparisons and ablations that would quantify the contribution of the backbone and the training components have not been run and are left to future work (Section 11); Table 8 compares only the headline figures with those published by Chakraborty (2025) [1].

It is important to emphasize that the proposed gap is primarily concerned with reproducibility, controlled evaluation, and end-to-end deep learning classification, rather than solving the complete real-world receipt-forgery problem. The current study evaluates synthetic splice and re-rendering manipulations on numeric-field crops. Therefore, detection of real-world forged receipts, whole-receipt localization, handwritten alterations, printed-over modifications, and previously unseen manipulation techniques remain open research problems for future investigation.

3. Dataset and Preprocessing

3.1 Source data

Receipt images and box annotations come from the SROIE ICDAR 2019 data [6] as distributed at <https://www.kaggle.com/datasets/priyanshumukherjee/bill-data/data>. The code scans the dataset root for annotation files (one line per box: four corner coordinates followed by text) and pairs each file with the image of the same file stem.

3.2 Crop selection

A box is kept if its text contains at least one digit, its width is at least 40 pixels and its height is at least 12 pixels. Receipts with fewer than two such boxes are skipped. For each remaining receipt, up to 16 boxes (default `per_receipt=16`) are sampled without replacement. Each box is cropped with 4 pixels padding.

3.3 Synthetic forgery generation

For every sampled box one authentic and one forged crop are created, so classes are balanced by construction. A donor crop is drawn from a different numeric box of the same receipt. With probability 0.4 the forgery is a splice: the donor, resized to $(w-8) \times (h-8)$, is pasted at offset (4,4). Otherwise it is a re-rendering: the interior of the crop is filled with the mean color of its four corner pixels, at least $\lfloor n_d/2 \rfloor$ digits (n_d = number of digits, minimum one) are each replaced by a different random digit, and the string is drawn at (6,3) in gray (intensity 10–70) with the default Python Imaging Library (PIL) font of size $\max(8, \lfloor 0.7h \rfloor)$ (Table 3).

3.4 Shared compression, resizing and splitting

Both authentic and forged crops are JPEG-compressed with quality drawn uniformly from $\{60, \dots, 95\}$, which prevents compression itself from being a class cue. Crops are resized to $H \times W = 112 \times 320$ with bicubic interpolation and stored as uint8 tensors. Samples are grouped by receipt, receipts are shuffled with the run seed, and split 70%/15%/15% into training, validation and test sets, so no receipt contributes to more than one split. Table 2 lists the resulting sizes; the training and test counts are recovered from the confusion matrices in Figures 2 and 3.

Table 2: Dataset Composition

Split	Samples	Authentic	Forged
Train	21,586	10,793	10,793
Test	4,602	2,301	2,301

Table 3: Forgery Synthesis Settings

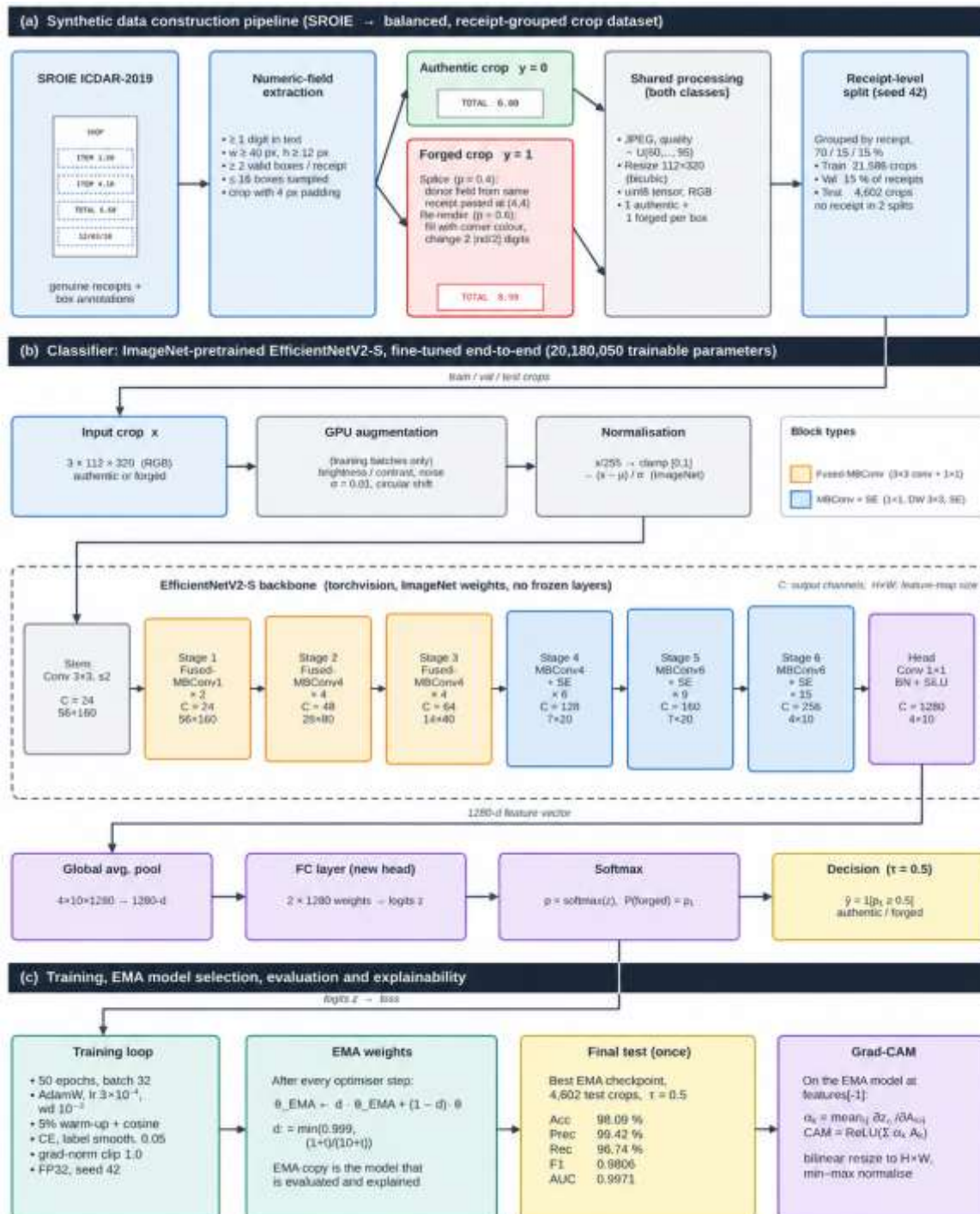
Parameter	Value
Crops per receipt (max)	16 (each yields 1 authentic + 1 forged)
Box filter	≥ 1 digit, $w \geq 40$ pixels, $h \geq 12$ pixels
Crop padding	4 pixels
Splice probability	0.4 (donor from same receipt)
Re-render digit change	$\geq \lfloor n_d/2 \rfloor$ digits, min 1
Text color / font	gray 10–70 / PIL default, size $\max(8, \lfloor 0.7h \rfloor)$
JPEG quality (both classes)	uniform integer 60–95
Input size	112×320, bicubic
Split	70/15/15 by receipt, seeded

4. Proposed Methodology

The proposed methodology is designed as a controlled and reproducible pipeline for detecting forged numeric fields in receipt images. Since the SROIE dataset provides genuine receipt images and text-box annotations but does not contain labeled forged samples, the proposed framework first constructs paired authentic and synthetically forged numeric-field crops. These samples are then subjected to a common compression and resizing procedure before being classified using an ImageNet-pretrained EfficientNetV2-S network. The complete workflow consists of numeric-field extraction, synthetic forgery generation, preprocessing, receipt-level dataset splitting, end-to-end model fine-tuning, exponential moving average (EMA) weight updating, validation-based model selection, and final test evaluation. Grad-CAM is

subsequently used to qualitatively examine the regions contributing to the model predictions. The overall workflow is illustrated in Figure 1.

Figure 1: Overall Framework: (a) Synthetic Authentic/Forged Crop Construction from SROIE with Receipt-Level Splitting; (b) EfficientNetV2-S Backbone and Two-Class Classification Head; (c) Training with EMA Weights, Test Evaluation and Grad-CAM.



Gradient updates flow from the loss (computed on logits z in panel b) to all backbone and head parameters; the EMA copy is never trained directly.
Forgeries are synthetic (splice / re-render); results quantify detection of this tempering process on SROIE, single seed.

4.1 Numeric-Field Extraction and Dataset Construction

The proposed system uses receipt images and corresponding text-box annotations from the SROIE dataset. For each receipt, the annotation files are paired with their corresponding receipt images using the common file stem. Only text boxes containing at least one numerical digit are considered because the proposed task focuses specifically on numeric fields that may contain amounts, dates, identifiers, or other numerical information. To remove very small or unsuitable regions, a numeric box is retained only when its width is at least 40 pixels and its height is at least 12 pixels. Receipts containing fewer than two valid numeric boxes are excluded. For each remaining receipt, a maximum of 16 numeric boxes is sampled without replacement, and each selected box is cropped with a padding of 4 pixels around the original bounding region.

For every selected numeric field, one authentic crop and one corresponding forged crop are generated. This procedure produces a balanced binary classification dataset in which the authentic class is assigned label ($y=0$) and the forged class is assigned label ($y=1$). The balanced construction avoids an artificial class-frequency difference between authentic and forged samples during training and evaluation.

4.2 Synthetic Forgery Generation

Because SROIE does not provide labeled forged versions of its receipt fields, two controlled synthetic manipulation mechanisms are used to generate forged samples: donor-field splicing and digit re-rendering. For the splicing operation, a donor numeric crop is selected from a different numeric box belonging to the same receipt. The donor crop is resized to fit the target field and pasted into the target crop with a fixed interior offset. This operation simulates a local replacement in which the numerical content of one receipt field is substituted with another numerical region. The probability of selecting the splicing mechanism is 0.4.

For the re-rendering operation, the interior of the target crop is first filled using the mean color obtained from its four corner pixels. At least half of the digits in the original numeric string are then replaced by different randomly selected digits, with at least one digit changed. The resulting string is rendered at a predefined position using a gray color and the default PIL font, with the font size determined from the crop height. This produces a forged field whose numerical content differs from the original while retaining the approximate spatial structure of the target field.

The two mechanisms are intentionally controlled and reproducible. However, they represent synthetic tampering processes rather than naturally occurring receipt manipulations; therefore, the proposed methodology evaluates the detection of the specified synthetic forgery processes rather than claiming direct generalization to real-world forged receipts.

4.3 JPEG Compression, Resizing, and Receipt-Level Data Splitting

After authentic and forged crops are generated, both classes undergo the same JPEG-compression procedure. The JPEG quality factor is randomly selected as an integer from 60 to 95. Applying the same compression process to both classes reduces the possibility that JPEG quality itself becomes a direct class-specific shortcut. The compressed crops are subsequently resized to a fixed spatial resolution of 112×320 pixels using bicubic interpolation and represented as RGB image tensors.

To prevent information leakage between data subsets, samples are grouped according to their source receipt before splitting. The receipts are shuffled using the experimental seed and divided into training, validation, and test subsets using a 70%/15%/15% split. Consequently, numeric crops originating from the same receipt cannot simultaneously occur in different subsets. This receipt-level splitting provides a

controlled evaluation setting and reduces the possibility that receipt-specific visual characteristics are shared between training and test samples.

4.4 EfficientNetV2-S Backbone and Classification Head

The reference classifier employs the EfficientNetV2-S architecture [7], implemented using the torchvision model library and initialized with its default ImageNet-pretrained weights to leverage transferable visual representations learned from large-scale image data. The complete network is fine-tuned end-to-end on the receipt numeric-field dataset, allowing the pretrained feature representations to adapt to the visual characteristics of authentic and synthetically forged receipt fields. No backbone layers are frozen during training; therefore, all network parameters, including the newly introduced classification head, are optimized jointly.

For binary forgery classification, the original final classification layer of EfficientNetV2-S is replaced with a fully connected layer containing two output units corresponding to the authentic and forged classes. The backbone produces a 1280-dimensional feature representation, which is passed to the classification head. The resulting classifier weight matrix has dimensions of 2×1280 , and the complete network contains 20,180,050 trainable parameters.

Each input crop is represented as a three-channel RGB image and resized to 112×320 pixels before being supplied to the network. Pixel values are first scaled to the range $[0,1]$ and subsequently normalized using the standard ImageNet channel-wise mean and standard deviation. The normalized images are then processed by EfficientNetV2-S to classify each receipt numeric-field crop as authentic or forged.

4.5 Training Strategy and Data Augmentation

The model is trained for exactly 50 epochs using the AdamW optimizer [8] with a learning rate of 3×10^{-4} , weight decay of 10^{-2} , and a batch size of 32. Cross-entropy loss with label smoothing of 0.05 [11, 12] is used to improve generalization and reduce overconfident predictions. To stabilize optimization, the gradient L2-norm is clipped at 1.0 [14]. The learning rate follows a two-stage schedule consisting of a 5% linear warm-up followed by cosine decay [9, 10], with updates performed at every training iteration. Mixed-precision training is disabled in the reported configuration to ensure reproducibility.

Training batches are generated using a seeded random generator, and the final incomplete batch is discarded. The experimental seed controls the Python, NumPy, and PyTorch random generators and enables deterministic CUDA Deep Neural Network library (cuDNN) and PyTorch algorithm settings wherever supported. Operations without available deterministic implementations generate warnings, and the CUDA Basic Linear Algebra Subroutines (cuBLAS) workspace configuration is explicitly set to support deterministic execution. These settings improve reproducibility across repeated runs under the same software and hardware environment.

Data augmentation is applied exclusively to training batches on the graphics processing unit (GPU). For each sample, brightness and contrast factors are independently sampled as $1 + 0.3(u - 0.5)$, where $u \sim U(0,1)$, yielding perturbation factors in the range $[0.85, 1.15]$. Additive Gaussian noise with a standard deviation of $\sigma = 0.01$ is then applied. In addition, a circular spatial shift is applied to each batch, with vertical displacement $dy \in [-4, 4]$ pixels and horizontal displacement $dx \in [-8, 8]$ pixels, sampled once per batch and shared across its samples. Finally, the augmented values are clipped to the valid range $[0, 1]$. Model selection is performed using validation accuracy only, ensuring that the validation set is used exclusively for evaluation and selection rather than training or gradient updates.

4.6 Exponential Moving Average and Model Selection

To obtain a more stable representation of the evolving network parameters, an exponential moving average (EMA) of the model weights is maintained throughout training [13]. The EMA model is updated after every optimizer step using a decay parameter of 0.999, together with the specified warm-up formulation. After each training epoch, the EMA model is evaluated on the validation set using a fixed decision threshold of 0.5. The checkpoint achieving the highest validation accuracy is retained, with ties resolved in favor of the later epoch. The best-validation EMA checkpoint is used for checkpoint storage, final test evaluation, and explainability analysis. Test-set metrics are monitored and recorded for plotting purposes throughout training, including periodic reporting every 10 epochs and epoch-wise tracking; however, they are never used for model selection. Although validation-based threshold tuning is supported as an option, it is not used in the reported configuration. The final reported test performance is obtained by loading the best-validation EMA checkpoint and evaluating it once on the held-out test set using the fixed threshold of 0.5.

Table 4: Model and Training Configuration

Item	Value
Backbone	EfficientNetV2-S, 2-class head
Input	3×112×320
Optimizer	AdamW, learning rate 3e-4, weight decay 1e-2
Schedule	5% warm-up, cosine, per step
Epochs / batch size	50 / 32
Loss	Cross-entropy, label smoothing 0.05
Gradient clipping	L2 norm 1.0
EMA decay	0.999 (with warm-up)
Precision	32-bit floating point (mixed precision off)
Threshold	0.5 (fixed)
Seed	42 (single run)
Trainable parameters	20,180,050

4.7 Overall Processing Pipeline

The complete proposed methodology can therefore be summarized as follows. First, numeric text boxes are extracted from SROIE receipts using the specified digit, width, and height criteria. Second, each selected field is paired with an authentic crop and a synthetically forged crop generated through either donor-field splicing or digit re-rendering. Third, both classes undergo identical JPEG compression and resizing to 112 × 320 pixels. Fourth, receipt-level grouping is used to create the 70% training, 15% validation, and 15% test partitions. Fifth, an ImageNet-pretrained EfficientNetV2-S model with a two-class head is fine-tuned end-to-end for 50 epochs using AdamW, label smoothing, gradient clipping, warm-up/cosine learning-rate scheduling, and training-only augmentation. Sixth, EMA weights are updated throughout optimization and the best EMA checkpoint is selected exclusively according to validation accuracy. Finally, the selected model is evaluated on the held-out test set at a fixed threshold of 0.5, while Grad-CAM is used to provide qualitative insight into the model's decision regions.

The resulting methodology provides a fully specified experimental pipeline for studying numeric-field forgery detection under the particular synthetic tampering processes defined in this work. Its conclusions are consequently limited to the controlled dataset and manipulation procedures described above.

5. Mathematical Formulation and Algorithms

5.1 Input Normalization and Classification Head

Let $x \in \{0, \dots, 255\}^{3 \times H \times W}$ denote an input RGB receipt-field crop, where H and W represent the image height and width, respectively. The corresponding ground-truth label is denoted by $y \in \{0, 1\}$, where $y = 0$ represents an authentic field and $y = 1$ represents a forged field. Each input image is first scaled to the range $[0, 1]$ and subsequently standardized using the ImageNet channel-wise mean $\mu = (0.485, 0.456, 0.406)$ and standard deviation $\sigma = (0.229, 0.224, 0.225)$. The normalized input image is obtained as

$$\tilde{x} = \frac{x/255 - \mu}{\sigma}. \quad (1)$$

The normalized image $\tilde{x}$ is passed through the EfficientNetV2-S-based detector, parameterized by θ , which produces a two-dimensional logit vector $z = (z_0, z_1)$ corresponding to the authentic and forged classes:

$$z = f_{\theta}(\tilde{x}) \in \mathbb{R}^2. \quad (2)$$

The logits are converted into class probabilities using the softmax function:

$$p_k = \frac{\exp(z_k)}{\sum_{j=0}^1 \exp(z_j)}, \quad k \in \{0, 1\}. \quad (3)$$

Here, $p_0 = P(y = 0 | x)$ represents the predicted probability of an authentic field, while $p_1 = P(y = 1 | x)$ represents the predicted probability of a forged field. The final predicted label $\hat{y}$ is determined using a fixed classification threshold $\tau = 0.5$:

$$\hat{y} = \begin{cases} 1, & \text{if } p_1 \geq \tau \\ 0, & \text{if } p_1 < \tau \end{cases} \quad \tau = 0.5. \quad (4)$$

5.2 Loss Function and Learning-Rate Schedule

To reduce prediction overconfidence and improve generalization, label smoothing is applied during training with a smoothing coefficient $\varepsilon = 0.05$ and $K = 2$ classes. The smoothed target probability distribution q is defined as

$$q_k = (1 - \varepsilon) \mathbb{1}[k = y] + \frac{\varepsilon}{K}, \quad (5)$$

where $\mathbb{1}[k = y]$ is an indicator function that equals 1 when k corresponds to the ground-truth class and 0 otherwise. Accordingly, the target probability assigned to the true class is $1 - \varepsilon + \varepsilon/K = 0.975$, while the remaining class receives a probability of $\varepsilon/K = 0.025$. The classification loss is computed using label-smoothed cross-entropy:

$$\mathcal{L} = - \sum_{k=0}^{K-1} q_k \log p_k. \quad (6)$$

For a mini-batch containing B training samples, the average classification loss is expressed as

$$\mathcal{L}_{\text{batch}} = -\frac{1}{B} \sum_{i=1}^B \sum_{k=0}^{K-1} q_{i,k} \log p_{i,k}, \quad (7)$$

where $q_{i,k}$ and $p_{i,k}$ denote the smoothed target and predicted probabilities, respectively, for the i -th sample and k -th class.

The learning rate follows a linear warm-up strategy during the initial 5% of the total optimization steps, followed by cosine decay. Let S denote the total number of optimizer steps, and let W represent the number of warm-up steps, calculated as

$$W = \max(1, \lfloor 0.05S \rfloor). \quad (8)$$

The learning rate at optimization step s is defined as

$$\eta_s = \begin{cases} \eta_0 \frac{s+1}{W}, & \text{for } 0 \leq s < W, \\ \frac{\eta_0}{2} \left[1 + \cos \left(\pi \frac{s-W}{S-W} \right) \right], & \text{for } W \leq s < S, \end{cases} \quad (9)$$

where η_0 denotes the base learning rate and s represents the current optimization-step index. This scheduling strategy gradually increases the learning rate during warm-up and subsequently reduces it through cosine annealing to support stable optimization and convergence. The formulation assumes $S > W$, with indexing consistent with the scheduler implementation.

5.3 Exponential Moving Average

To improve the stability of model parameters during training, an exponential moving average (EMA) of the network weights is maintained. Let θ_t represent the current network parameters after optimization step t , and let $\bar{\theta}_t$ denote the corresponding EMA parameters. The step-dependent EMA decay coefficient is defined as

$$d_t = \min \left(\delta, \frac{1+t}{10+t} \right), \quad (10)$$

where $\delta = 0.999$ denotes the maximum EMA decay coefficient. The EMA parameters are updated after each optimization step according to

$$\bar{\theta}_t = d_t \bar{\theta}_{t-1} + (1 - d_t) \theta_t. \quad (11)$$

The EMA parameters are initialized from the initial network weights as

$$\bar{\theta}_0 = \theta_0. \quad (12)$$

In this formulation, the decay coefficient d_t controls the contribution of previously accumulated parameters relative to the current network parameters. Non-floating-point buffers are copied directly from the current network rather than exponentially averaged. By smoothing parameter fluctuations across optimization steps, EMA provides a more stable set of model weights for validation and final evaluation. The precise EMA step indexing and buffer-update behavior are implementation-dependent and should be consistent with the training code.

Together, these mathematical operations define the input preprocessing, probabilistic prediction, loss computation, learning-rate scheduling, and EMA-based parameter averaging employed in the EfficientNetV2-S-based receipt forgery detector.

5.4 Grad-CAM

To interpret the visual regions contributing to the classifier's predictions, Gradient-weighted Class Activation Mapping (Grad-CAM) [15] is applied to the exponential moving average (EMA) model at the final stage of the EfficientNetV2-S convolutional feature extractor (features[-1]). Grad-CAM generates class-discriminative activation maps by combining convolutional feature maps with gradient-based importance weights associated with the predicted class. This approach enables qualitative examination of the image regions influencing the model's classification of authentic and forged receipt fields.

Let $A^k \in \mathbb{R}^{h \times w}$ denote the k -th feature map extracted from the selected convolutional layer, where h and w represent its spatial dimensions. For a predicted class $c \in \{0,1\}$, the corresponding class logit is denoted by z_c . The importance weight α_k^c for feature map A^k is computed by globally averaging the gradients of the class logit with respect to the spatial activations:

$$\alpha_k^c = \frac{1}{hw} \sum_{i=1}^h \sum_{j=1}^w \frac{\partial z_c}{\partial A_{ij}^k}. \quad (13)$$

Here, A_{ij}^k represents the activation at spatial position (i, j) in the k -th feature map, and $\partial z_c / \partial A_{ij}^k$ represents the sensitivity of the class logit to that activation. The resulting importance weights indicate the relative contribution of individual feature maps to the predicted class.

The class-discriminative Grad-CAM activation map is subsequently obtained as

$$G_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_{k=1}^C \alpha_k^c A^k \right), \quad (14)$$

where C denotes the total number of feature maps in the selected convolutional layer. The rectified linear unit (ReLU) retains positive contributions to the target-class logit while suppressing negative activation values, highlighting spatial regions that positively influence the classification decision.

Since the Grad-CAM activation map is initially generated at the lower spatial resolution of the selected convolutional layer, it is resized to match the original input crop dimensions $H \times W$ using bilinear interpolation:

$$G_{\text{resized}}^c = \text{Interp}_{H \times W}(G_{\text{Grad-CAM}}^c), \quad (15)$$

where $\text{Interp}_{H \times W}(\cdot)$ denotes bilinear interpolation to the original image height and width.

The resized activation map is then normalized to the range $[0,1]$ using min-max normalization:

$$\hat{G}_{ij}^c = \frac{G_{\text{resized},ij}^c - \min(G_{\text{resized}}^c)}{\max(G_{\text{resized}}^c) - \min(G_{\text{resized}}^c) + \epsilon_0}, \quad (16)$$

where ϵ_0 is a small positive constant introduced to prevent division by zero when the activation map has little or no variation. The normalized activation map $\hat{G}^c$ provides a spatial representation of the relative importance of image regions for the predicted class.

For qualitative visualization, the normalized Grad-CAM heatmap can be superimposed on the original receipt-field image:

$$I_{\text{overlay}} = (1 - \lambda) I_{\text{original}} + \lambda \text{ColorMap}(\hat{G}^c), \quad (17)$$

where I_{original} denotes the original RGB image scaled to $[0,1]$, $\text{ColorMap}(\cdot)$ converts the normalized activation map into a three-channel heatmap, and $\lambda \in [0,1]$ controls the transparency of the heatmap overlay.

The resulting visualizations are used to qualitatively investigate whether the classifier focuses on numerical characters, altered digit boundaries, local text structures, background patterns, or other visual regions when predicting authentic and forged receipt fields. By examining the spatial distribution of model attention, Grad-CAM provides insight into the visual evidence associated with individual predictions. However, the highlighted regions represent gradient-based attribution rather than definitive proof of image manipulation, and their interpretation remains qualitative.

5.5 Algorithms

Algorithm 1 builds the paired authentic/forged crops for each receipt and splits them by receipt; Algorithm 2 trains the detector and selects the best EMA checkpoint on validation accuracy only.

Algorithm 1: Synthetic pair generation for one receipt		
Input: receipt image I , filtered numeric boxes B , samples per receipt n , random generator r		
1	if $ B < 2$ then skip receipt	guard
2	for each b in $r.\text{sample}(B, \min(n, B))$ do	
3	donor $\leftarrow$ crop(I , random box $\neq b$); real $\leftarrow$ crop(I , b)	
4	if $r.\text{random}() < 0.4$ then	
5	forged $\leftarrow$ paste(real, resize(donor), (4, 4))	splice forgery
6	else	
7	fill interior of real with mean corner color; replace $\geq \lfloor n_d/2 \rfloor$ digits of text(b) by different digits; draw text at (6, 3)	erase and re-render
8	end if	
9	add (to_tensor(JPEG(real)), 0) and (to_tensor(JPEG(forged)), 1) to the receipt group	
10	end for	
11	shuffle groups with seed; split 70/15/15 by group	receipt-grouped split

Algorithm 2: Training, EMA and model selection		
Input: train/val/test sets, seed, epochs $E = 50$, batch size 32		
1	set seeds and deterministic flags; build model, EMA copy, AdamW and schedule (9)	
2	for epoch = 1 ... E do	
3	for each shuffled batch (x , y) do	
4	$\tilde{x} \leftarrow$ normalize(augment(x)) (1); compute loss (7); backpropagate	forward / backward
5	clip gradient norm to 1; AdamW step; schedule step (9); EMA update (11)	update
6	end for	
7	evaluate EMA model on validation set ($\tau = 0.5$)	

8	if val accuracy $\geq$ best then store EMA weights as best	model selection
9	record read-only test curves (plotting only); every 10th epoch print test metrics	
10	end for	
11	load best EMA weights; report final test metrics at $\tau = 0.5$	

6. Experimental Setup

The framework is PyTorch [17] with torchvision models. Training was conducted on a single NVIDIA Tesla T4 GPU using PyTorch 2.11.0+cu130 and CUDA 13.0. The total training time was 4 h 13 min, which corresponds to an average of about 304 seconds per epoch. The code prints the GPU name and PyTorch version and notes that results are comparable only on the same GPU type and library version. Only seed 42 was reported. The training code fixes the random seeds and enables deterministic flags, so this run can be reproduced on the same GPU type and library versions.

7. Evaluation Metrics

Using true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) with forged as the positive class:

Table 5: Evaluation Metrics and Their Formulas

Metric	Formula	Eq.
Accuracy	$\text{Acc} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$	(18)
Precision	$\text{Prec} = \text{TP} / (\text{TP} + \text{FP})$	(19)
Recall	$\text{Rec} = \text{TP} / (\text{TP} + \text{FN})$	(20)
F1-score	$\text{F1} = 2 \cdot \text{Prec} \cdot \text{Rec} / (\text{Prec} + \text{Rec})$	(21)
AUC	$\text{AUC} = [\sum_{i: y_i=1} \text{rank}(p_i) - P(P+1)/2] / (P \cdot N)$	(22)

Eq. (22) is the rank-sum (Mann–Whitney) form of the area under the receiver operating characteristic (ROC) curve (AUC) [16], with average ranks for ties, P forged and N authentic samples. Specificity or true-negative rate (TNR) and false-positive rate (FPR=1–TNR) are derived from the confusion matrix. Precision, recall and the F1-score (the harmonic mean of precision and recall) are reported for the forged class.

8. Results and Analysis

8.1 Final test performance

Table 6 reports the final result of the best-validation EMA model on the test set (4,602 crops) at $\tau=0.5$, as logged. The confusion matrix is shown in Figure 2 (TN=2,288, FP=13, FN=75, TP=2,226). Recomputing from these counts gives 98.09% accuracy, 99.42% precision, 96.74% recall and F1 0.9806, matching the printed values; the AUC of 0.9971 (0.99710 in the saved `results_seed42.json`) is taken from the logged output, not recomputed from counts. The ROC curve is shown in Figure 4: it rises almost vertically along the left axis and reaches roughly 0.99 true-positive rate by a false-positive rate of about 0.05 (read from the figure), consistent with the AUC of 0.9971.

Table 6: Final Test Results (Seed 42, $\tau = 0.5$)

Accuracy	Precision	Recall	F1	AUC
98.09% (0.98088)	99.42% (0.99419)	96.74% (0.96741)	0.9806	0.9971

Values from the saved results_seed42.json (arch efficientnet_v2_s, threshold 0.5); rounded values match the console output.

Figure 2: Confusion Matrix on the Test Set at Threshold 0.5.

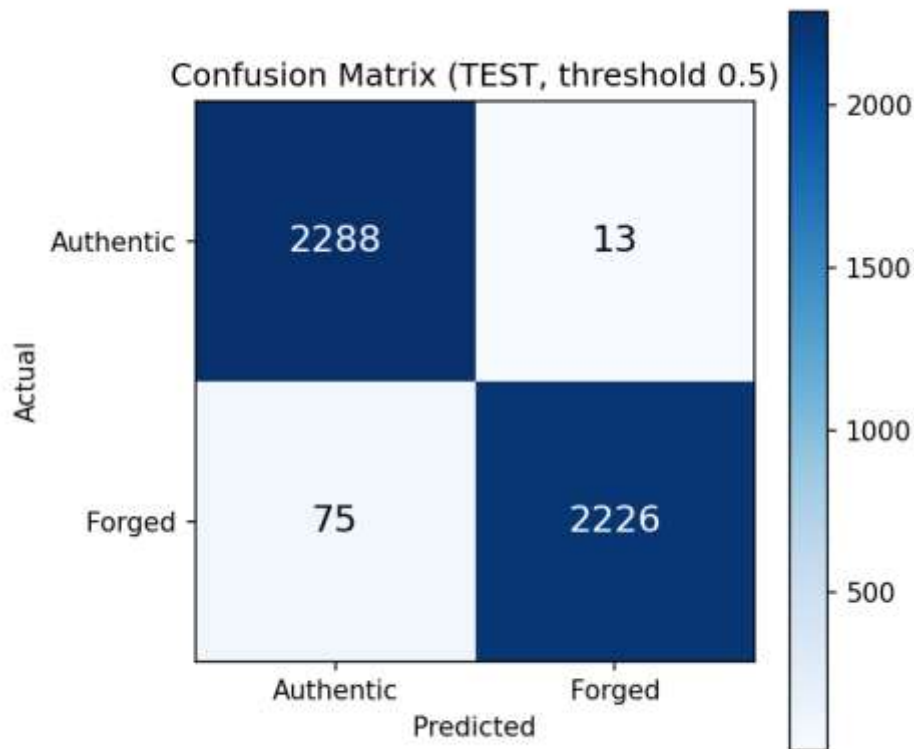


Table 7: Confusion Matrices and Derived Rates

Set	TN	FP	FN	TP	Acc	Prec	Rec
Train	10,785	8	61	10,732	99.68%	99.93%	99.43%
Test	2,288	13	75	2,226	98.09%	99.42%	96.74%

Test FPR = $13/2,301 = 0.57\%$; test false-negative rate (FNR) = $75/2,301 = 3.26\%$. Train metrics are derived here from Figure 3.

Figure 3: Confusion Matrix of the Selected Model on the Training Set at Threshold 0.5.

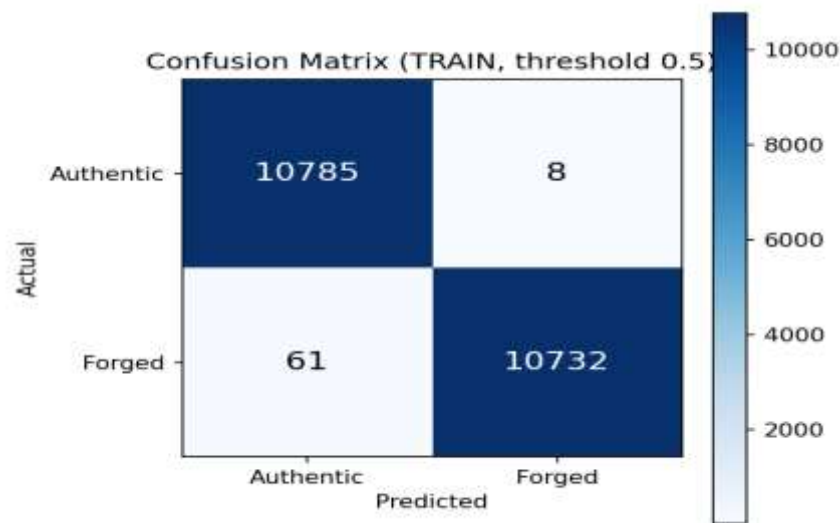
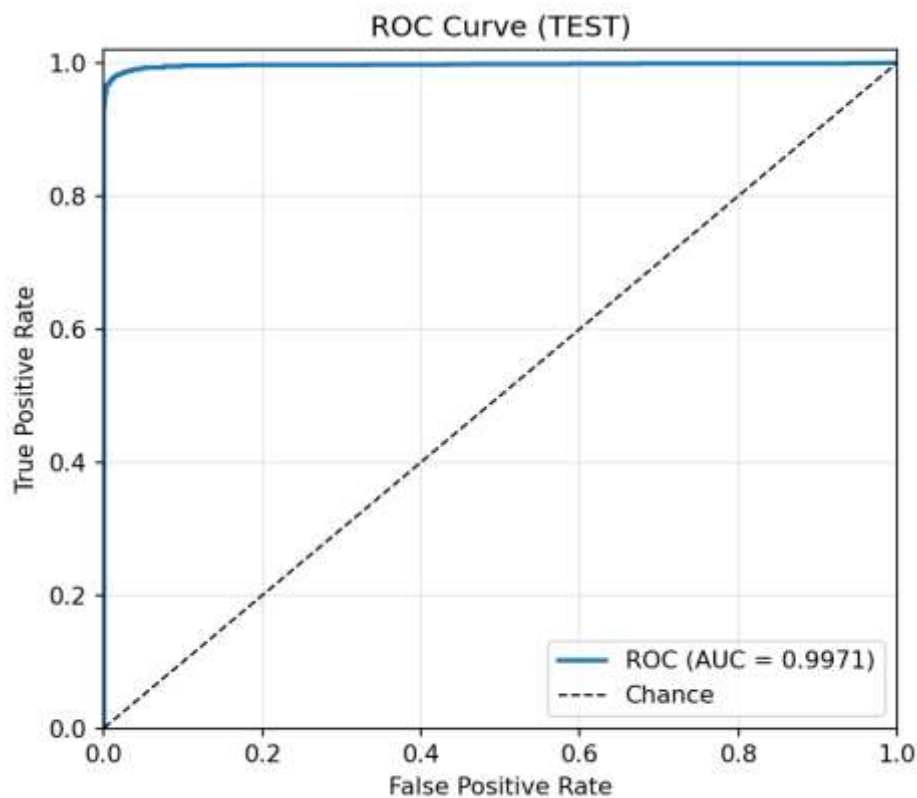


Figure 4: ROC Curve on the Test Set (AUC = 0.9971).



8.2 Learning curves

Figure 5 shows the training loss (left), which decreases monotonically to about 0.12, and training accuracy against validation accuracy of the EMA model (right). Validation accuracy rises quickly, reaches its highest values of about 0.985 between roughly epochs 17 and 26, and then declines slowly to about 0.980 by epoch 50 while training accuracy approaches 1.0 (all read from the figure). Additional training after

the validation peak therefore did not improve held-out accuracy, which is why validation-based checkpoint selection matters here.

Figure 5: Left: Training Loss. Right: Training Accuracy and Validation Accuracy (EMA Model).

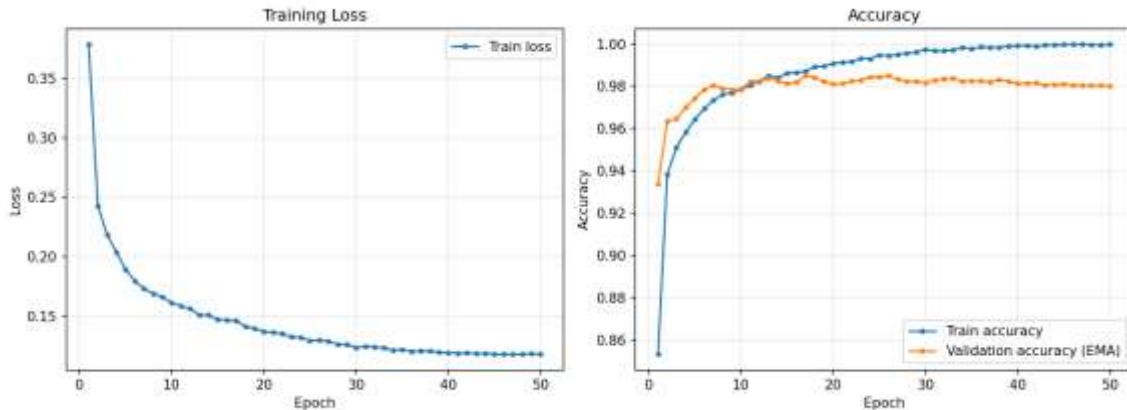
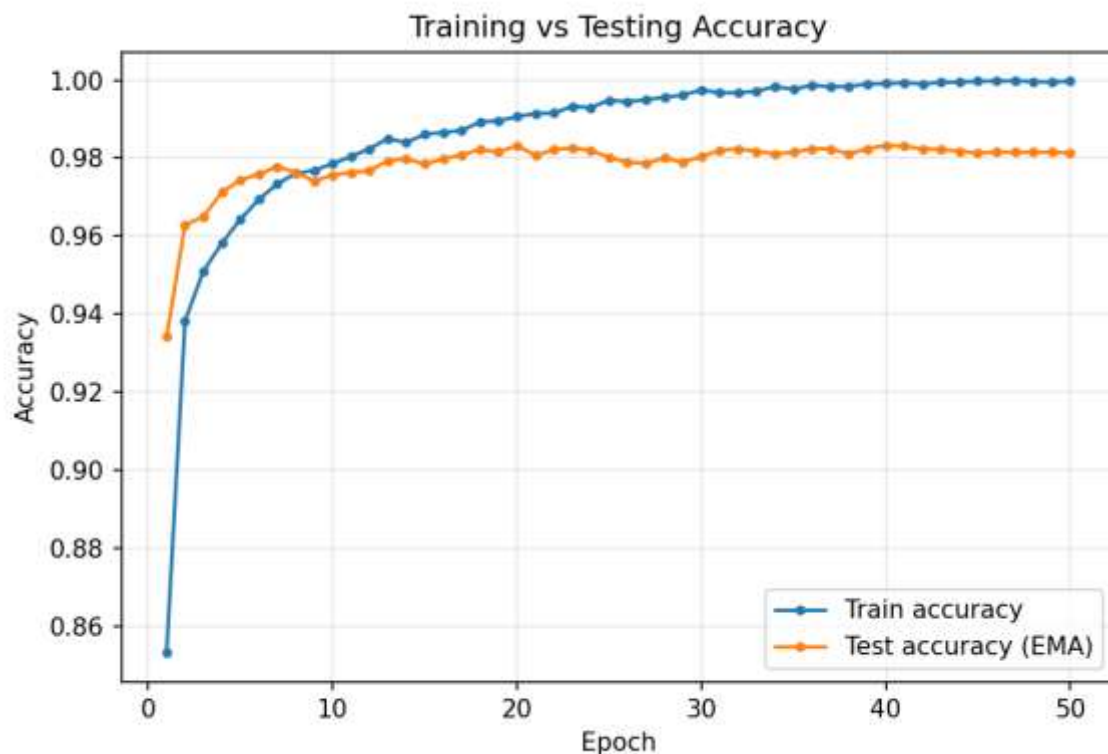


Figure 6 shows training accuracy rising to nearly 1.0 while EMA test accuracy plateaus at roughly 0.98 after about 20 epochs (values read from the figure). The training accuracy in this figure is accumulated during each epoch in training mode with augmentation on the non-EMA weights, whereas the test curve uses the EMA model in evaluation mode, so the two are not measured under identical conditions. The evaluation-mode training accuracy of the selected model is 99.68% (Table 7).

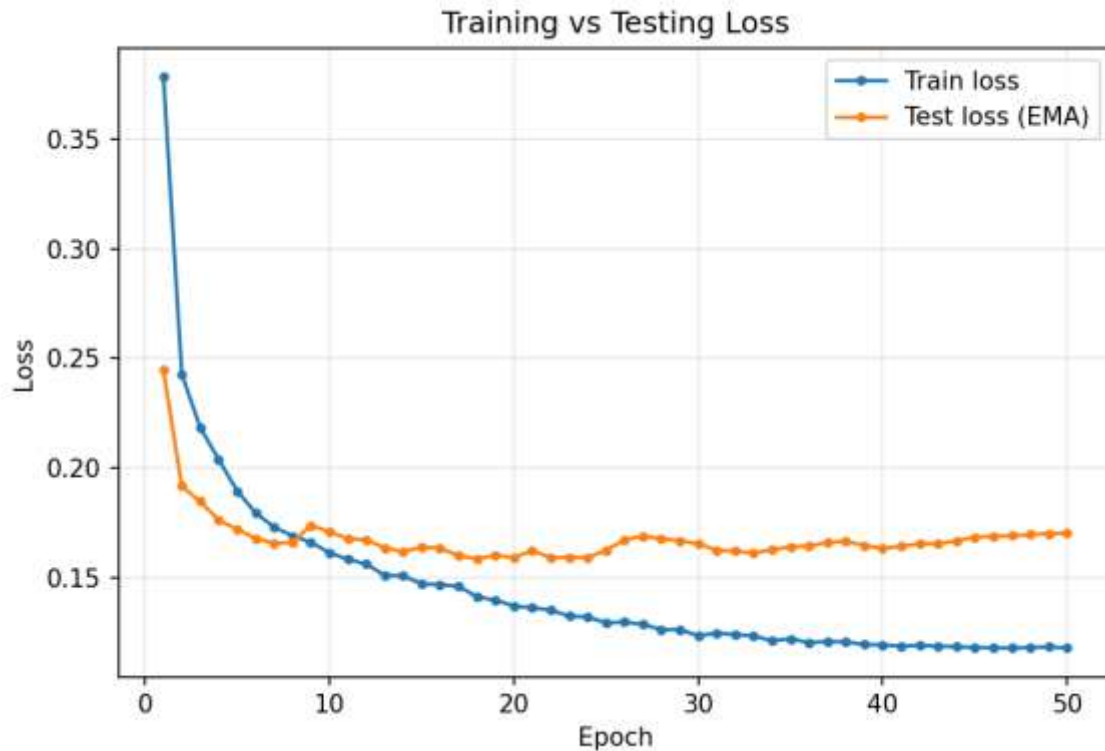
Figure 6: Training Accuracy and EMA Test Accuracy over 50 Epochs.



In Figure 7 the training loss falls steadily to about 0.12, whereas the test loss reaches a minimum near 0.16 around epochs 15–25 and then drifts up slightly to about 0.17 by epoch 50 (read from the figure). Both losses include label smoothing (0.05), so neither can approach zero. A widening gap with flat test accuracy

indicates mild overfitting or increasing confidence on misclassified samples; it was not investigated further.

Figure 7: Training Loss and EMA Test Loss (Both with Label Smoothing 0.05).



8.3 Precision, recall and F1 across epochs

Figure 8 presents the validation and test precision of the EMA model over the 50 training epochs, while Figures 9–11 show the corresponding test precision, recall, and F1-score, respectively. The test precision remains consistently high throughout training, varying approximately between 98.8% and 99.7%. It reaches its highest value near epoch 16 and remains close to 99% toward the end of training. In comparison, validation precision varies approximately between 97.4% and 98.9% and ends at around 97.8%, indicating a small difference between validation and test precision.

The test recall improves substantially during the initial training stages. It increases from approximately 87.9% at epoch 1 to around 96–97% between epochs 15 and 20. The recall subsequently continues to fluctuate, reaching a maximum of approximately 97.5% around epoch 33 before ending at approximately 97.1% at epoch 50. This behavior indicates that the model progressively becomes more effective at identifying forged receipt fields as training proceeds.

The test F1-score also improves rapidly during the early epochs, increasing from approximately 93.0% at epoch 1 to about 98.3% around epochs 20 and 40. It finishes at approximately 98.1% at epoch 50. The consistently high F1-score demonstrates a favorable balance between precision and recall for the forged class.

It is important to note that the epoch-50 values observed in the plots do not correspond exactly to the final metrics reported in Table 6. The final reported test precision, recall, and F1-score are 99.42%, 96.74%, and 0.9806, respectively. The plotted values around epoch 17 are approximately 99.4% precision, 96.7% recall, and 98.06% F1-score, which closely match the final reported results. This suggests that the selected best-validation checkpoint was likely obtained around epoch 17 rather than at epoch 50.

Figure 8: Validation and Test Precision (EMA) over Epochs.

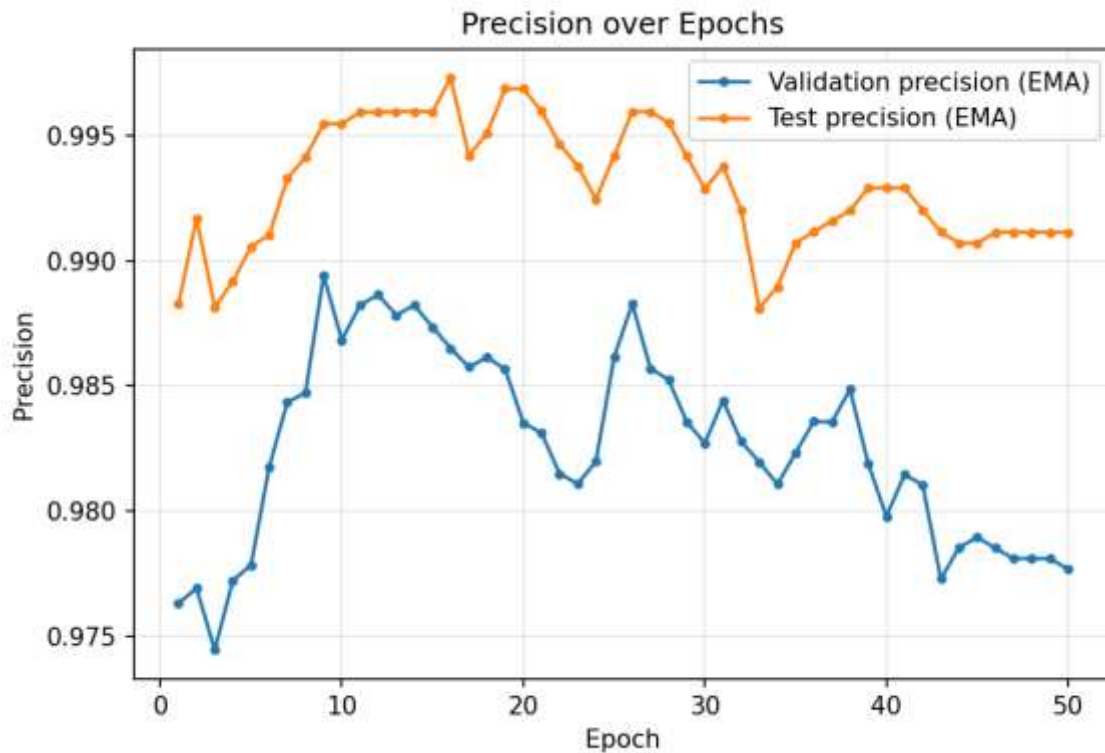


Figure 9: Test Precision (%) versus Number of Epochs.

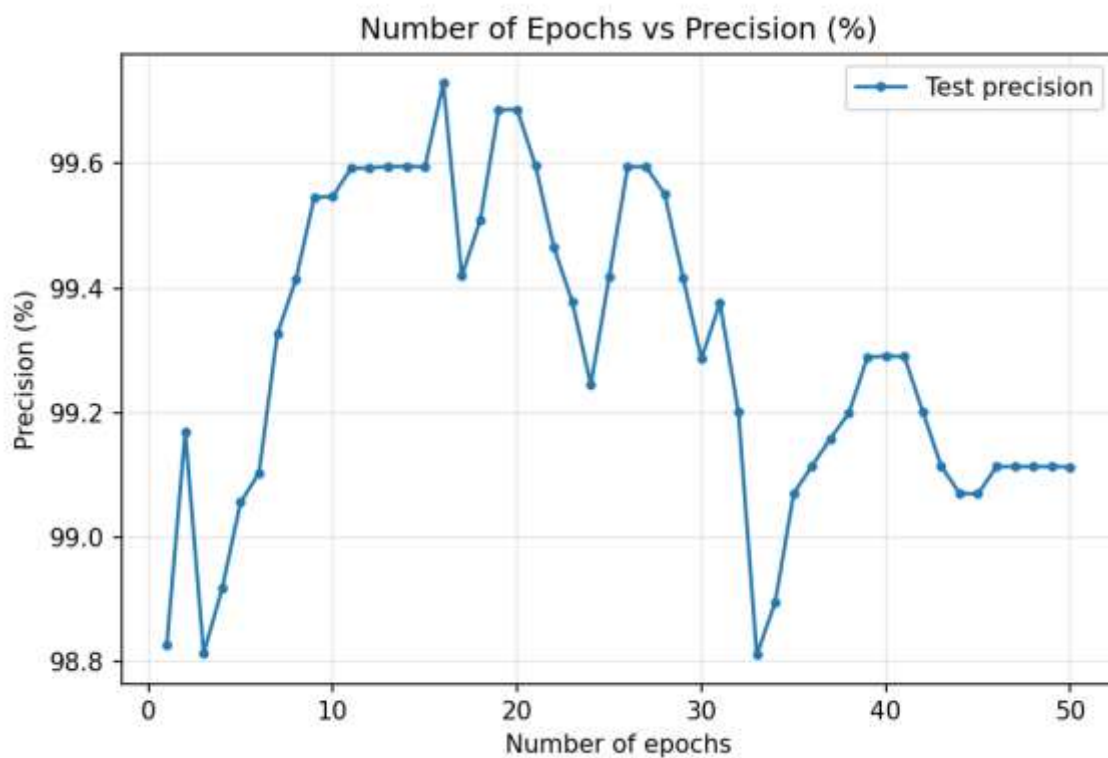


Figure 10: Test Recall (%) versus Number of Epochs.

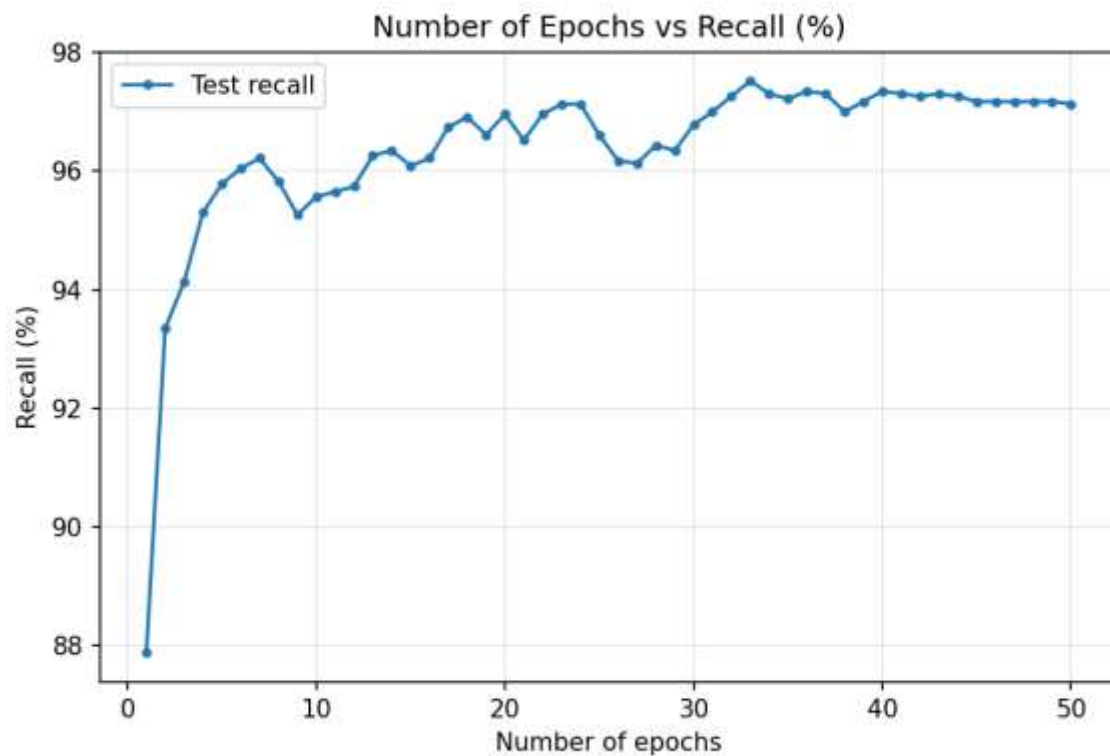
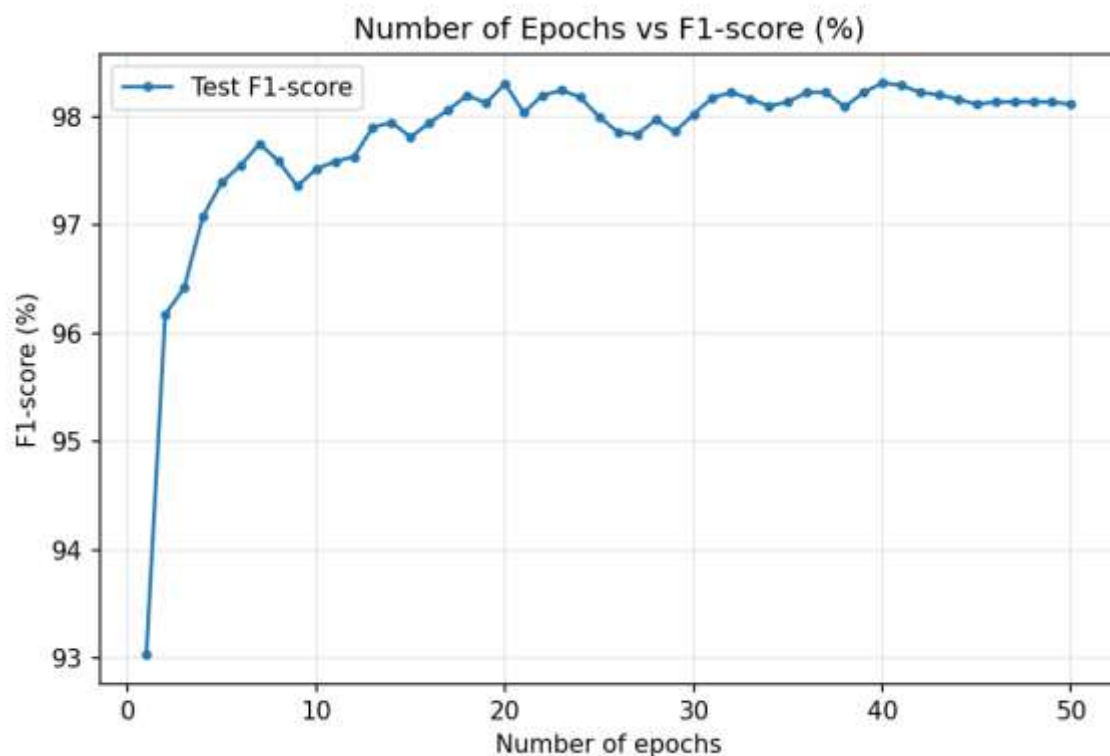


Figure 11: Test F1-Score (%) versus Number of Epochs.

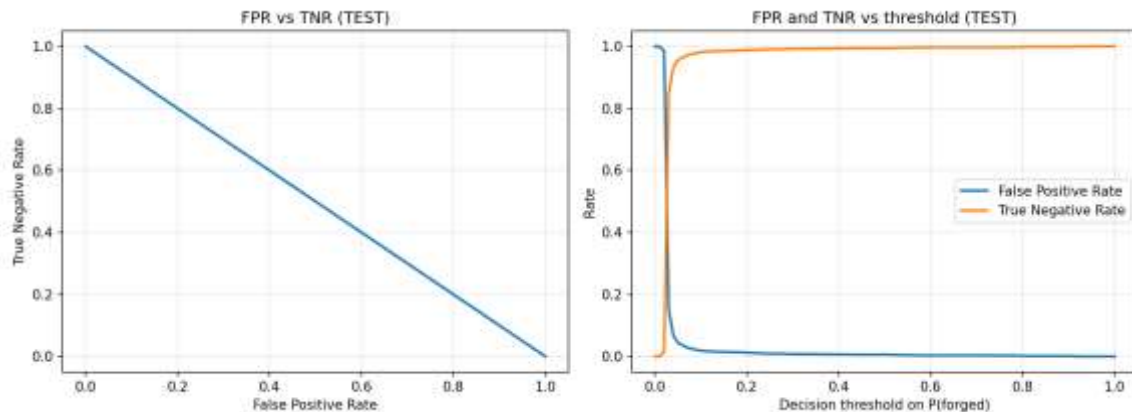


8.4 Threshold behavior

Figure 12 (right) shows FPR and TNR on the test set as the threshold on $P(\text{forged})$ is swept. FPR is close to 1 for thresholds below about 0.02 and falls steeply between about 0.02 and 0.05, after which it stays

near zero. The left panel plots TNR against FPR; since $TNR=1-FPR$ by definition, it is a straight line that carries no information, and the ROC curve in Figure 4 is the informative alternative. The abrupt drop is consistent with the label-smoothing target of 0.025 for the non-true class, which bounds confident predictions near $P(\text{forged}) \approx 0.025$ (the Grad-CAM panels in Figure 13 show 0.02–0.03 for three of the four authentic samples, 0.16 for the fourth, and 0.98 for all four forged samples). This is an interpretation, not a tested result.

Figure 12: Left: TNR versus FPR (Test). Right: FPR and TNR versus Decision Threshold.



8.5 Comparison with previous work and scope of baselines

This work is compared only with Chakraborty (2025) [1], using the figures that study publishes. No baseline was re-run, because Chakraborty (2025) [1] uses a private dataset.

Table 8: Comparison with the Results Reported by Chakraborty (2025)

Study	Test data	Test size	Acc	Prec	Rec	F1	AUC
This work (EfficientNetV2-S)	Public SROIE only; numeric-field crops, synthetic forgeries	4,602	98.09%	99.42%	96.74%	0.9806	0.9971
Chakraborty (2025) [1] (as published)	SROIE + private dataset	202	97.1%	97.3%	97.5%	0.974†	—
Difference (this work – Chakraborty (2025))	—	—	+0.99 pp	+2.12 pp	–0.76 pp	+0.007†	—

† Derived from the precision and recall reported by Chakraborty (2025) [1]. — : not reported or not applicable. pp = percentage points.

The present model has higher accuracy and precision but lower recall (Table 8). This is not a ranking: the test sets differ in size (4,602 versus 202), data (public SROIE versus SROIE plus a private set), task (numeric-field crops versus receipt regions) and protocol, and the 95% accuracy intervals (about 97.7–98.4% versus 93.8–98.7%) overlap.

No baselines, ablations or repeated seeds were run; these are left to future work (Section 11).

9. Discussion

The experimental results demonstrate that the EfficientNetV2-S-based reference classifier can effectively distinguish authentic and synthetically tampered numeric fields in SROIE receipts. At the fixed decision threshold of 0.5, the selected EMA model achieves 98.09% accuracy, 99.42% precision, 96.74% recall,

an F1-score of 0.9806, and an AUC of 0.9971 on the held-out test set. The high precision indicates that most samples classified as forged are indeed forged, while the slightly lower recall shows that a small proportion of forged fields remain undetected. The corresponding confusion matrix contains 13 false positives and 75 false negatives, indicating that missed forgeries are more frequent than false alarms.

The difference between precision and recall is particularly relevant for a fraud-screening application. Although the model produces only a small number of false alarms, the larger number of false negatives indicates that some forged numeric fields are visually similar to authentic fields and are therefore more difficult to detect. At the current threshold, the test false-positive rate is 0.57%, whereas the false-negative rate is 3.26%. Consequently, if the primary objective is to reduce missed forgeries, a lower decision threshold may be worth investigating. However, the appropriate threshold should be selected using the validation set rather than the test set. Although the threshold-sweep analysis shows that the false-positive rate remains low across part of the threshold range, corresponding recall values for each threshold were not recorded in the results; therefore, a specific alternative threshold cannot be recommended from the current experiment alone.

The learning curves also indicate a measurable generalization gap. Training accuracy approaches 100%, whereas the validation and test performance remain close to 98%. In addition, the training loss continues to decrease while the test loss reaches its minimum approximately between epochs 15 and 25 and then increases slightly toward the end of training. This behavior suggests mild overfitting or increasing model confidence on incorrectly classified samples. Importantly, the data were divided at the receipt level, so crops originating from the same receipt were not distributed across the training, validation, and test sets. Therefore, the observed generalization gap cannot be attributed to direct receipt-level overlap between the data subsets.

The results should nevertheless be interpreted within the controlled nature of the synthetic forgery-generation procedure. The forged samples are generated using two predefined mechanisms: donor-field splicing and digit re-rendering. The re-rendered text uses a default font and a simplified background reconstruction, while spliced fields involve resized donor crops. These operations can introduce visual characteristics that differ systematically from authentic receipt regions. Although the same JPEG-compression procedure is applied to authentic and forged samples to reduce compression quality as a class-specific shortcut, other synthesis-related artifacts remain. Consequently, the model may learn some characteristics of the synthetic generation process rather than general visual evidence of real-world receipt manipulation. The reported performance should therefore be interpreted as detection capability for the specific synthetic tampering procedures used in this study, rather than as evidence of equivalent performance on naturally occurring forged receipts.

The Grad-CAM analysis provides additional qualitative information about the learned representations. For the selected examples, the model frequently activates around the numeric text line and, in some cases, near crop boundaries or background regions. These observations are consistent with the possibility that the classifier uses character-level, boundary, and rendering-related cues when distinguishing authentic and forged fields. However, only eight examples were examined qualitatively, and no quantitative faithfulness or occlusion analysis was performed. Therefore, the Grad-CAM results should be regarded as qualitative evidence rather than definitive proof of the specific visual features used by the model.

Compared with the reference study by Chakraborty (2025) [1], which reports 97.1% accuracy, 97.3% precision, and 97.5% recall, the present model obtains higher reported accuracy and precision but lower recall (96.74% versus 97.5%). However, a direct performance ranking is not justified because the two

studies use different datasets, forgery-generation procedures, test-set sizes, and evaluation protocols. The present study uses 4,602 test crops generated from publicly accessible SROIE receipts, whereas the reference study uses 202 samples from SROIE and a private dataset. Thus, the comparison is provided only as contextual evidence rather than as a claim of superiority. Furthermore, the present study does not include baseline architectures, ablation experiments, or repeated-seed evaluation, which limits conclusions regarding the individual contribution of EfficientNetV2-S, EMA, label smoothing, augmentation, and other training components.

Overall, the findings support the effectiveness of the proposed pipeline for the controlled detection of the synthetic numeric-field forgeries considered in this study. At the same time, the single-seed evaluation, absence of real forged receipts, lack of ablation studies, and restricted focus on numeric-field crops limit the strength of broader conclusions. Further evaluation on real forged receipts, additional manipulation types, multiple random seeds, and controlled baseline and ablation experiments is necessary to determine whether the observed performance generalizes beyond the synthetic tampering process used here.

10. Explainability and Error Analysis

To investigate the visual evidence used by the proposed EfficientNetV2-S classifier, Grad-CAM was applied to the EMA model at the final convolutional feature stage. Eight test samples were selected using a seeded random procedure, consisting of four authentic and four forged numeric-field crops. As shown in Figure 13, all eight selected samples were correctly classified. The authentic samples produced forged-class probabilities $P(\text{forged})$ ranging from 0.02 to 0.16, whereas the forged samples produced probabilities of approximately 0.98. These examples therefore demonstrate confident predictions for both classes, although the limited number of visualized samples does not permit general conclusions about the complete test set.

The Grad-CAM maps provide qualitative evidence regarding the spatial regions associated with the model predictions. For the forged samples, the strongest activation generally appears as a horizontal band along the numeric text line. This behavior is visible for examples containing fields such as “RM 80.70” and “23 MAR 2517”. One forged example also exhibits a relatively strong rectangular activation region associated with the local background area. These patterns indicate that the model is responding to visual information distributed around the numeric field rather than necessarily relying only on individual digit shapes.

For several authentic samples, the strongest Grad-CAM activation is instead concentrated close to the crop boundaries, including regions around the final character or the left edge of the crop. The difference between the authentic and forged examples suggests that the classifier may exploit a combination of character, boundary, background, and rendering-related visual cues. However, this interpretation remains qualitative. Grad-CAM identifies regions associated with a model prediction but does not by itself establish that the highlighted regions are causally responsible for the classification decision.

The observed activation patterns should also be interpreted in relation to the synthetic forgery-generation process used in this study. The forged samples are produced using two mechanisms: donor-field splicing and digit re-rendering. In the splicing procedure, a resized donor numeric crop is inserted into the target field, whereas the re-rendering procedure reconstructs the local background and replaces at least half of the digits with different randomly selected digits. Such operations can introduce differences in local boundaries, background appearance, text rendering, or spatial structure. Therefore, the activation observed around text lines and boundary/background regions may be consistent with the model exploiting cues

introduced by the synthetic generation process. Nevertheless, the current analysis does not establish whether these cues are the primary basis of the model's decision.

The quantitative error analysis provides a complementary view of the classifier's behavior. On the held-out test set of 4,602 crops, the selected model produced 13 false positives and 75 false negatives at the fixed decision threshold of 0.5. Thus, the false-positive rate among authentic crops was $(13/2301=0.57\%)$, while the false-negative rate among forged crops was $(75/2301=3.26\%)$. The corresponding training-set confusion matrix contains 8 false positives and 61 false negatives. False negatives therefore constitute the dominant error type in both the training and test sets.

The larger number of false negatives indicates that a subset of forged numeric fields remains difficult to distinguish from authentic fields. This observation is consistent with the slightly lower recall of 96.74% compared with the precision of 99.42% reported for the final test model. However, the supplied experiment does not record the forgery mechanism, digit count, or crop-quality characteristics for each misclassified sample. Consequently, it is not possible to determine from the present results whether false negatives are more strongly associated with donor-field splicing, digit re-rendering, shorter or longer numeric strings, or particular crop-quality conditions. Such an attribution would require an additional error-level analysis. The explainability results should also be considered together with the controlled nature of the dataset. Both authentic and forged samples undergo the same JPEG-compression procedure, reducing compression quality as a direct class-specific cue. Nevertheless, the synthetic operations themselves may introduce other visual characteristics that are not representative of naturally occurring receipt manipulations. The Grad-CAM observations are therefore useful for examining the behavior of the trained classifier within the present experimental setting, but they should not be interpreted as evidence that the model has learned general-purpose forensic features applicable to real-world receipt forgeries.

Figure 13: Grad-CAM on Test Crops (Top Four Authentic, Bottom Four Forged); Red Indicates the Most Influential Regions.



Overall, Figure 13 provides qualitative evidence that the reference detector attends to the numeric text region and, in some cases, to crop boundaries and local background regions. The quantitative error analysis further shows that the remaining errors are dominated by false negatives, indicating that some synthetic forgeries are visually close to their authentic counterparts. Because the Grad-CAM analysis is based on only eight selected examples and no quantitative faithfulness or occlusion test was performed, the findings should be regarded as qualitative rather than definitive. A stronger explainability and error analysis would require a larger systematically selected sample set, quantitative explanation-faithfulness evaluation, and separate analysis of errors according to forgery mechanism, digit count, and crop characteristics.

11. Limitations and Future Work

Several limitations should be acknowledged. First, the evaluated forgeries are entirely synthetic and comprise only two forgery types; therefore, no real-world tampered receipts were included in the evaluation. Second, the proposed method classifies crops corresponding to numeric fields rather than complete receipt images, and it does not perform forgery localization. Third, the reported results are based on a single random seed and a single data split, and thus the variability of the results across different splits or random initializations has not been quantified. Fourth, baseline comparisons and ablation studies have not yet been conducted. Fifth, the validation-set size and the exact best-validation epoch are not reported. Finally, the evaluation does not include whole-receipt images, handwritten modifications, printed-over edits, or AI-generated forgeries.

Future work will address these limitations by evaluating the proposed approach on real forged-receipt datasets, such as the dataset reported by Tornés, et al. (2023) [2]. Additional forgery types that do not share the rendering characteristics or signatures of the training data will also be investigated to assess the model's generalization capability. Furthermore, experiments will be repeated across multiple random seeds to quantify performance variability and improve the statistical reliability of the results. Finally, the baseline comparisons (alternative backbones) and ablation studies (removing EMA, label smoothing and augmentation) will be completed to provide a more comprehensive assessment of the contribution of each component of the proposed method.

12. Conclusion

This study showed that numeric-field forgery detection can be examined under a controlled protocol built entirely from public SROIE data, in which every forged crop has an authentic counterpart, both classes share the same compression, and receipts are kept within a single data split. Under this protocol, the EfficientNetV2-S reference detector separated authentic and forged crops with high reliability, and its remaining errors were mostly missed forgeries rather than false alarms. Qualitative inspection suggested that the predictions may rest partly on cues introduced by the synthesis process, so the results describe the specific synthetic tampering considered here and not real-world receipt manipulation. Evaluation on real forged receipts, additional manipulation types, alternative backbones, ablation studies and repeated runs is the necessary next step to establish how far the approach generalizes.

13. References

1. D. Chakraborty, "Bill safe: Intelligent forgery detection with CNN and upgrade sand cat swarm optimization", Discover Computing, 2025, 28, article 84. <https://doi.org/10.1007/s10791-025-09602-5>

2. B.M. Tornés, T. Taburet, E. Boros, K. Rouis, P. Gomez-Krämer, N. Sidere, V.P. d'Andecy, "Receipt dataset for document forgery detection", International Conference on Document Analysis and Recognition (ICDAR), Lecture Notes in Computer Science, Springer, 2023, 14189, 454–469. https://doi.org/10.1007/978-3-031-41682-8_28
3. B.M. Tornés, E. Boros, A. Doucet, P. Gomez-Krämer, J.-M. Ogier, "Detecting forged receipts with domain-specific ontology-based entities & relations", International Conference on Document Analysis and Recognition (ICDAR), Lecture Notes in Computer Science, Springer, 2023, 14189, 184–199. https://doi.org/10.1007/978-3-031-41682-8_12
4. L. Zhao, C. Chen, J. Huang, "Deep learning-based forgery attack on document images", IEEE Transactions on Image Processing, 2021, 30, 7964–7979. <https://doi.org/10.1109/TIP.2021.3112048>
5. R. Sukhija, M. Kumar, M.K. Jindal, "Document forgery detection: A comprehensive review", International Journal of Data Science and Analytics, 2025, 20, 4385–4407. <https://doi.org/10.1007/s41060-025-00723-0>
6. Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, C.V. Jawahar, "ICDAR2019 competition on scanned receipt OCR and information extraction", International Conference on Document Analysis and Recognition (ICDAR), 2019, 1516–1520. <https://doi.org/10.1109/ICDAR.2019.00244>
7. M. Tan, Q.V. Le, "EfficientNetV2: Smaller models and faster training", International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, 2021, 139, 10096–10106. <https://doi.org/10.48550/arXiv.2104.00298>
8. I. Loshchilov, F. Hutter, "Decoupled weight decay regularization", International Conference on Learning Representations (ICLR), 2019. <https://doi.org/10.48550/arXiv.1711.05101>
9. I. Loshchilov, F. Hutter, "SGDR: Stochastic gradient descent with warm restarts", International Conference on Learning Representations (ICLR), 2017. <https://doi.org/10.48550/arXiv.1608.03983>
10. P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour", arXiv, 2017. <https://doi.org/10.48550/arXiv.1706.02677>
11. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the Inception architecture for computer vision", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>
12. R. Müller, S. Kornblith, G. Hinton, "When does label smoothing help?", Advances in Neural Information Processing Systems (NeurIPS), 2019, 32. <https://doi.org/10.48550/arXiv.1906.02629>
13. B.T. Polyak, A.B. Juditsky, "Acceleration of stochastic approximation by averaging", SIAM Journal on Control and Optimization, 1992, 30 (4), 838–855. <https://doi.org/10.1137/0330046>
14. R. Pascanu, T. Mikolov, Y. Bengio, "On the difficulty of training recurrent neural networks", International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, 2013, 28 (3), 1310–1318. <https://doi.org/10.48550/arXiv.1211.5063>
15. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization", IEEE International Conference on Computer Vision (ICCV), 2017, 618–626. <https://doi.org/10.1109/ICCV.2017.74>
16. J.A. Hanley, B.J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve", Radiology, 1982, 143 (1), 29–36. <https://doi.org/10.1148/radiology.143.1.7063747>
17. A. Paszke, et al., "PyTorch: An imperative style, high-performance deep learning library", Advances in Neural Information Processing Systems (NeurIPS), 2019, 32, 8024–8035.

<https://doi.org/10.48550/arXiv.1912.01703>

18. B. Bayar, M.C. Stamm, “A deep learning approach to universal image manipulation detection using a new convolutional layer”, ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), 2016, 5–10. <https://doi.org/10.1145/2909827.2930786>
19. Y. Rao, J. Ni, “A deep learning approach to detection of splicing and copy-move forgeries in images”, IEEE International Workshop on Information Forensics and Security (WIFS), 2016, 1–6. <https://doi.org/10.1109/WIFS.2016.7823911>
20. P. Zhou, X. Han, V.I. Morariu, L.S. Davis, “Learning rich features for image manipulation detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 1053–1061. <https://doi.org/10.1109/CVPR.2018.00116>
21. Y. Wu, W. AbdAlmageed, P. Natarajan, “ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 9543–9552. <https://doi.org/10.1109/CVPR.2019.00977>
22. L. Verdoliva, “Media forensics and deepfakes: An overview”, IEEE Journal of Selected Topics in Signal Processing, 2020, 14 (5), 910–932. <https://doi.org/10.1109/JSTSP.2020.3002101>
23. C. Qu, C. Liu, Y. Liu, X. Chen, D. Peng, F. Guo, L. Jin, “Towards robust tampered text detection in document image: New dataset and new solution”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, 5937–5946. <https://doi.org/10.1109/CVPR52729.2023.00575>
24. D. Luo, et al., “Toward real text manipulation detection: New dataset and new solution”, arXiv:2312.06934, 2023. <https://doi.org/10.48550/arXiv.2312.06934>
25. A. Gupta, A. Vedaldi, A. Zisserman, “Synthetic data for text localisation in natural images”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 2315–2324. <https://doi.org/10.1109/CVPR.2016.254>
26. R. Geirhos, et al., “Shortcut learning in deep neural networks”, Nature Machine Intelligence, 2020, 2, 665–673. <https://doi.org/10.1038/s42256-020-00257-z>
27. S. Kornblith, J. Shlens, Q.V. Le, “Do better ImageNet models transfer better?”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 2661–2671. <https://doi.org/10.1109/CVPR.2019.00277>
28. D. Cozzolino, L. Verdoliva, “Noiseprint: A CNN-based camera model fingerprint”, IEEE Transactions on Information Forensics and Security, 2020, 15, 144–159. <https://doi.org/10.1109/TIFS.2019.2916364>
29. M. Huh, A. Liu, A. Owens, A.A. Efros, “Fighting fake news: Image splice detection via learned self-consistency”, European Conference on Computer Vision (ECCV), 2018, 101–117. https://doi.org/10.1007/978-3-030-01252-6_7
30. J. Adebayo, J. Gilmer, M. Muehly, I. Goodfellow, M. Hardt, B. Kim, “Sanity checks for saliency maps”, Advances in Neural Information Processing Systems (NeurIPS), 2018, 31, 9505–9515. <https://doi.org/10.48550/arXiv.1810.03292>