Transfer-Learned Convolutional–Recurrent Networks for Generalizable Video Deepfake Detection Under Cross-Dataset Shift

***NIMMAGADDA PRASANTHI[1], MOTHA PRAMEELA[2]**

1PG Student, Department of CSE, VIKAS COLLEGE OF ENGINEERING AND TECHNOLOGY, Nunna, Vijayawada Rural, A.P
2 Assistant Professor, Department of CSE, VIKAS COLLEGE OF ENGINEERING AND TECHNOLOGY, Nunna, Vijayawada Rural, A.P
*Corresponding author: santhi.kanakavalli@gmail.com

Abstract

Deepfake videos produced by autoencoder- and GAN-based face-swapping pipelines are increasingly realistic, and detectors trained on one benchmark frequently degrade on videos generated under different conditions. The present study examines two factors that govern the generalisability of learning-based video deepfake detectors: ImageNet transfer learning with staged fine-tuning, and explicit temporal modelling of frame sequences. An XceptionNet single-frame classifier is trained on face crops from DFD, Celeb-DF, DFDC-Preview and their union, and evaluated under a complete self- and cross-dataset protocol that includes a custom test set of self-generated FaceSwap-GAN and in-the-wild deepfakes. The backbone is then extended into CNN, CNN-LSTM and CNN-GRU sequence classifiers evaluated on DFD, DFDC, FaceForensics++ and a merged custom dataset. Transfer learning improved accuracy in 18 of 20 train–test configurations (mean gain 4.97 percentage points), and training on the combined distribution raised accuracy on the custom test set from 63.19% to 73.20%. Nevertheless, single-source models lost 20–35 percentage points when evaluated on unseen datasets, confirming dataset shift as the dominant limitation. Among eight optimisers, Nadam yielded the highest accuracy for all three architectures. The CNN-LSTM model achieved the highest 70-epoch accuracy on every dataset (85.2% on DFD; precision 0.895, recall 0.797, F1-score 0.843), whereas CNN-GRU did not consistently outperform the frame-level baseline. A CNN-LSTM trained on pooled multi-source data with transfer learning reached 81.54% on unseen blended and in-the-wild samples. The findings indicate that source diversity and pre-trained initialisation contribute more to cross-domain robustness than recurrent modelling alone.

Keywords: *Deepfake detection; XceptionNet; transfer learning; convolutional–recurrent neural network; LSTM; GRU; dataset shift; cross-dataset generalisation*

1. INTRODUCTION

Deepfakes are synthetic image, video, or audio content generated using deep learning to alter a person's identity, expressions, or speech. Initially associated with face-swapping techniques based on autoencoders, deepfake generation has advanced significantly through generative adversarial networks (GANs) [1] and style-based generators [2]. The principal video manipulation techniques include identity swapping, which replaces a target's face with another person's face, and facial reenactment, which transfers expressions, head poses, and mouth movements [3]. Tools such as FaceSwap-GAN [4] enable realistic manipulations with limited computational resources. Although deepfakes have legitimate applications in filmmaking and visual effects, their misuse for misinformation, defamation, fraud, and biometric spoofing has increased the need for reliable automated detection [5].

Existing deepfake detectors primarily identify artefacts introduced during synthesis. Face warping and blending can create resolution inconsistencies [6], while colour discrepancies and boundary artefacts can be detected using convolutional neural networks (CNNs), including MesoNet [7] and XceptionNet [8]. XceptionNet has demonstrated strong frame-level performance on the FaceForensics++ benchmark [9]. However, analysing individual frames may overlook temporal inconsistencies, including flickering, landmark variations, and illumination changes. Consequently, hybrid architectures combining CNN-based feature extraction with recurrent neural networks (RNNs) have been investigated [10], [11]. Other approaches exploit physiological signals that generative models may reproduce inaccurately [12].

A major challenge is generalisation across datasets. Public benchmarks differ in manipulation quality, resolution, compression, generation methods, and subject characteristics [13], [14]. Models trained on one dataset may learn generator-specific artefacts and perform poorly on unseen data, a problem known as dataset shift [15], closely associated

with domain adaptation [16]. Therefore, high accuracy on a benchmark's test set does not necessarily indicate reliable performance on real-world videos.

Transfer learning and temporal modelling are potential approaches to improving generalisation. Transfer learning uses networks pre-trained on large datasets, such as ImageNet [17], and fine-tunes them for deepfake detection [18]. Such pre-training may provide general visual features and reduce dependence on generator-specific artefacts. Temporal modelling complements frame-level analysis by identifying inconsistencies across video sequences. Recurrent architectures, particularly long short-term memory (LSTM) [20] and gated recurrent unit (GRU) networks [21], can capture these temporal patterns [10], [19].

Nevertheless, the individual contributions of transfer learning, training-source diversity, temporal modelling, and optimiser selection to cross-dataset performance remain insufficiently established. Systematic comparisons involving different datasets, recurrent architectures, and optimisation strategies are needed to guide practical detector development. In particular, controlled comparisons between CNN, CNN–LSTM, and CNN–GRU models, together with evaluations on unseen and in-the-wild manipulations, remain important research directions [10], [22].

To address these gaps, this study investigates the generalisability of video deepfake detection within a unified experimental framework. Its main contributions are:

- Cross-dataset evaluation: Assessment of XceptionNet with and without ImageNet transfer learning across DFD, Celeb-DF, DFDC-Preview, their combined dataset, and a custom test set containing self-generated FaceSwap-GAN videos and in-the-wild deepfakes.
- Architectural comparison: Controlled evaluation of frame-level CNN, CNN–LSTM, and CNN–GRU models using a common XceptionNet backbone, four datasets, and three training budgets, with accuracy, precision, recall, and F1-score as evaluation metrics.
- Optimiser assessment: Empirical comparison of eight optimisers, identifying Nadam as the most effective within the specified training budget.
- Generalisation analysis: Investigation of whether a transfer-learned CNN–LSTM trained on pooled benchmark datasets generalises better to unseen blended and in-the-wild samples than single-source models.

2. LITERATURE REVIEW

2.1 Deepfake generation and benchmark datasets

Early facial manipulation relied on computer-graphics pipelines; Face2Face, for example, achieved real-time monocular reenactment by transferring expressions captured with an RGB camera from a source actor to a target [3]. The introduction of GANs [1] and, subsequently, style-based generators [2] enabled the synthesis of photorealistic faces with controllable attributes. Most face-swap deepfakes circulating online are produced with autoencoders that share an encoder between two identities and use identity-specific decoders; FaceSwap-GAN [4] augments this scheme with an adversarial loss, self-attention and mask-based blending.

2.2 Frame-level detection

Classical image forensics detected tampering from JPEG quantisation and double-compression traces [25] and from constrained convolutional layers that learn prediction-error filters rather than image content [26]. For face manipulations, Matern *et al.* [6] exploited visual artefacts around the eyes, teeth and facial contours, and Yang *et al.* [27] used inconsistent three-dimensional head poses. MesoNet [7] analysed mesoscopic features with a shallow network, whereas Rössler *et al.* [9] showed that an XceptionNet [8] classifier trained on face crops outperformed hand-crafted and shallow learned alternatives on FF++. Pan *et al.* [28] similarly used XceptionNet and MobileNet on FF++, and EfficientNet [29] has become a common backbone owing to its favourable accuracy–parameter trade-off. Many of these detectors are initialised with ImageNet weights [17], exploiting transfer learning [18]; however, the influence of such initialisation on cross-dataset performance has rarely been isolated.

2.3 Temporal and convolutional–recurrent detection

Long-term recurrent convolutional networks [19] established the paradigm of combining convolutional perception with recurrent sequence modelling. Güera and Delp [10] applied this idea to deepfakes by feeding CNN frame features to an LSTM [20], and Sabir *et al.* [11] reported a 4.55% improvement over the previous state of the art on FF++ using a recurrent convolutional strategy. Hashmi *et al.* [22] examined a Conv-LSTM hybrid on DFDC with facial-landmark and facial-movement cues, and Montserrat *et al.* [32] combined an MTCNN face detector, a CNN, an automatic face-weighting layer and a gated recurrent unit (GRU) [21], reporting a test accuracy of 91.88% on DFDC. Other temporal cues include eye-blinking frequency [33] and inter-frame optical flow [34]. Although GRUs have fewer parameters than LSTMs and often perform comparably in sequence modelling [21], direct comparisons of CNN-LSTM and CNN-GRU detectors under identical conditions are uncommon.

2.4 Generalisation and dataset shift

Learning theory indicates that the error of a classifier on a target domain is bounded by its source error plus a measure of divergence between the two domains [16], and dataset shift can arise from non-stationary data-generation environments or biased sample selection [15]. In deepfake detection, both conditions apply: each benchmark is produced with particular generation software, actors, cameras and compression settings. Das *et al.* [35] observed that detectors overfit because many samples originate from few actors, and reported log-loss reductions of 15.2–35.3% through dynamic face augmentation. Surveys [5] likewise identify cross-dataset generalisation as an open problem. Transfer learning and domain adaptation share the objective of applying knowledge acquired in one context to another; transfer-learning-based CNN architectures [36], [37] and convolutional–recurrent variants [38] have therefore been proposed to improve generalisation. The present study consolidates and extends these directions by evaluating both factors within a unified cross-dataset protocol.

3. MATERIALS AND METHODOLOGY

3.1 Overview of the experimental design

The methodology comprises three experiments. Experiment A evaluates a single-frame XceptionNet classifier, trained with and without transfer learning, under a full self- and cross-dataset protocol. Experiment B extends the same backbone into three sequence classifiers (CNN, CNN-LSTM and CNN-GRU), selects the optimiser, and compares the architectures on four datasets. Experiment C trains the CNN-LSTM model on individual and pooled sources to examine multi-domain generalisation on unseen samples. All experiments treat deepfake detection as a supervised binary classification task in which "fake" is the positive class.

3.2 Datasets

Experiment A. Three benchmark distributions were used: DFD [23], Celeb-DF [13] and DFDC-Preview [14]. DFD and Celeb-DF videos were divided into training, validation and test subsets in the ratio 70:15:15, whereas DFDC-Preview followed its official split. A combined distribution was formed by merging the corresponding splits of the three datasets. To assess generalisation beyond the benchmarks, a custom test set of 100 videos with an equal number of real and fake samples was compiled: 40 videos were prepared in-house under controlled conditions (authentic recordings and self-generated face swaps), and 60 were collected from YouTube, representing some of the most convincing publicly circulating deepfakes. The self-generated fakes were produced with FaceSwap-GAN [4]. For this purpose, face images (500–1000 per identity) were extracted from press-conference, entertainment and self-recorded clips, aligned to reference landmarks and supplemented with binary eye masks. The generator is a convolutional autoencoder whose encoder contains five convolutional layers and two self-attention blocks, and the discriminator comprises four convolutional layers operating on 64 × 64 RGB inputs. Decoders of paired identities were swapped to produce face swaps, yielding 20 good-quality deepfake videos. The custom set was used exclusively for testing. The resulting frame-level corpora are listed in Table 1, and the relative positions of the datasets in terms of visual and manipulation quality are illustrated in Fig. 1.

Table 1. Number of face images in the frame-level corpora used for single-frame detection (Experiment A). The combined test split also contains the custom test set.

Dataset	Train	Validation	Test
DFDC-Preview	187,080	36,298	36,302
Celeb-DF	173,517	36,629	37,785
DFD	155,019	32,889	33,187
Custom (test only)	–	–	32,502
Combined	515,616	105,816	139,776

Note: Combined train = sum of the three benchmark training splits; combined test = sum of the three benchmark test splits and the custom test set.

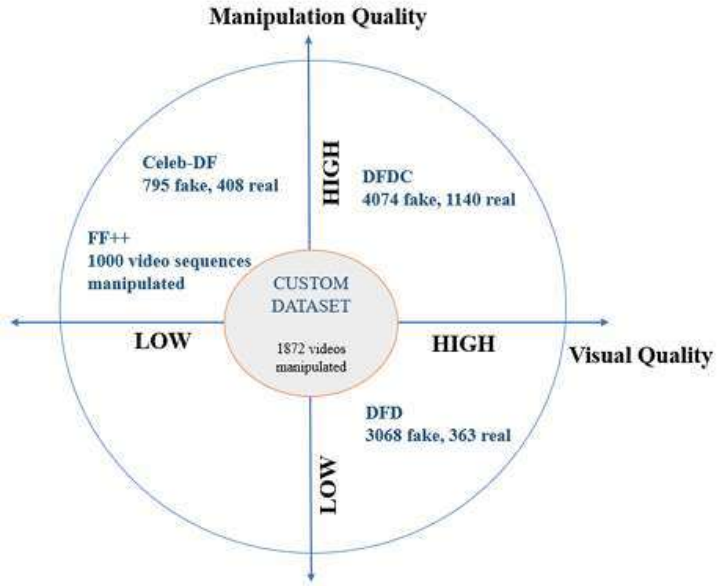


Fig. 1. Positioning of the benchmark and custom datasets according to visual quality and manipulation quality. Celeb-DF and DFD occupy opposite quadrants, whereas DFDC combines high visual and manipulation quality.

Experiments B and C. The sequence classifiers were trained and tested on DFD [23], DFDC [24], FF++ [9] and a custom dataset formed by merging samples from these benchmarks to provide a heterogeneous distribution. Real and fake videos of each dataset were divided into training, validation and test subsets in the ratio 70:15:15, with class balance maintained in every subset (Table 2).

Table 2. Video-level splits (real + fake) used for the sequence-classification experiments (Experiment B).

Dataset	Train (real + fake)	Validation (real + fake)	Test (real + fake)
DFD	328 + 328 = 656	70 + 70 = 140	70 + 70 = 140
DFDC	328 + 328 = 656	70 + 70 = 140	70 + 70 = 140
FF++	581 + 621 = 1202	125 + 133 = 258	125 + 133 = 258
Custom (merged)	984 + 984 = 1968	210 + 210 = 420	210 + 210 = 420

3.3 Pre-processing

Each video was decomposed into frames, and faces were localised using the CNN-based max-margin object detection (MMOD) face detector of the Dlib library [39], which returns a bounding rectangle around each face. Because authentic videos are considerably fewer than manipulated videos in all benchmarks, the frame-extraction rate was adjusted separately for each class so that the final corpora contained approximately equal numbers of real and fake images. A minimum face size was imposed to reject false detections and to avoid up-sampling small faces, since enlargement introduces interpolation artefacts that the classifier could misinterpret as manipulation traces. After experimentation with several square sizes, the threshold was fixed at 128 pixels, and all retained faces were resized to 128 × 128 pixels and labelled as real (0) or fake (1). The pipeline is illustrated in Fig. 2. Converting videos into face-centred image sequences reduces the dimensionality of the input, suppresses irrelevant background content and focuses learning on the facial region where manipulations occur.

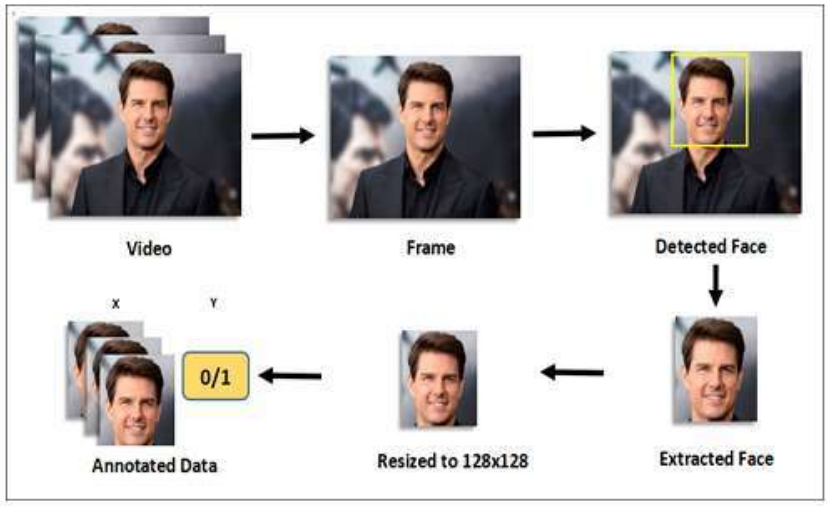


Fig. 2. Pre-processing pipeline: frame extraction, face detection, face cropping, resizing to 128 × 128 pixels and label assignment.

3.4 Single-frame detector and transfer-learning protocol

XceptionNet [8] was adopted as the feature extractor because of its reported effectiveness on FF++ [9]. It consists of 36 convolutional layers organised into 14 modules built from depthwise separable convolutions with linear residual connections, and its exit flow produces a 2048-dimensional descriptor through global average pooling. For an input face crop x, the descriptor is

$$v = \Phi_X(x; \theta), \quad v \in \mathbb{R}^{2048} \tag{1}$$

where Φ_X denotes the convolutional base with parameters θ. Two initialisation strategies were compared. Models without transfer learning were initialised with the method of He *et al.* [40] and trained for 30 epochs. Models with transfer learning were initialised with ImageNet weights [17]; their convolutional layers were frozen for the first 15 epochs so that only the classification head was trained, after which all layers were unfrozen and fine-tuned for a further 15 epochs, retaining the checkpoint with the highest validation accuracy. All models were trained with the Adam optimiser, a cyclical learning rate, a mini-batch size of 512 images and the binary cross-entropy loss

$$\mathcal{L}_{BCE} = -(1/N)\, \Sigma_{n=1}^{N} [\, y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \,] \tag{2}$$

where $y_n \in \{0, 1\}$ is the true label and $\hat{y}_n$ the predicted probability of the n-th sample. One model was trained on each of the four distributions (DFD, Celeb-DF, DFDC-Preview and combined) with and without transfer learning. Each model was tested on its own test split (self-test), on the test splits of the other distributions (cross-test) and on the custom test set. Because the combined test split is the union of the individual test splits, the accuracy of any model on the combined test set equals the sample-weighted average of its accuracies on the constituent sets:

$$Acc_{comb} = \Sigma_k N_k\, Acc_k \,/\, \Sigma_k N_k \tag{3}$$

where N_k is the number of test images in set k. Eq. (3) makes the combined accuracy a compact summary of self- and cross-set performance, and it was used to verify the internal consistency of the reported results.

3.5 Convolutional–recurrent sequence classifiers

The architecture used for sequence classification is shown in Fig. 3. A sequence of T consecutive face crops is processed by the ImageNet-initialised XceptionNet wrapped in a time-distributed layer, so that the same convolutional weights are applied to every frame and a 2048-dimensional vector v_t is obtained for each frame. A dropout layer with a rate of 0.5 follows the feature extractor to reduce overfitting. Three temporal heads were compared. In the frame-level CNN baseline, the per-frame features are aggregated by global average pooling and passed to a dense layer with sigmoid activation. In CNN-LSTM, the feature sequence is analysed by an LSTM layer [20] whose gates are defined as

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, v_t] + b_i), \quad f_t = \sigma(W_f[h_{t-1}, v_t] + b_f), \quad o_t = \sigma(W_o[h_{t-1}, v_t] + b_o) \\ \tilde{c}_t &= \tanh(W_c[h_{t-1}, v_t] + b_c), \quad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad h_t = o_t \odot \tanh(c_t) \end{aligned} \tag{4}$$

where i_t, f_t and o_t are the input, forget and output gates, c_t is the cell state, h_t the hidden state and $\odot$ the element-wise product. In CNN-GRU, the LSTM is replaced by a GRU [21], which merges the cell and hidden states and uses only update (z_t) and reset (r_t) gates:

$$z_t = \sigma(W_z[h_{t-1}, v_t] + b_z), \quad r_t = \sigma(W_r[h_{t-1}, v_t] + b_r)$$
$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, v_t] + b_h), \quad h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \qquad (5)$$

The probability that a sequence is fake is then obtained from the pooled features for the CNN baseline and from the final hidden state for the recurrent models:

$$\hat{y}_{CNN} = \sigma(w^\top (1/T) \Sigma_{t=1}^{T} v_t + b), \qquad \hat{y}_{RNN} = \sigma(w^\top h_T + b) \qquad (6)$$

A sequence is classified as fake when $\hat{y}$ exceeds 0.5. The inference procedure therefore consists of four steps: extraction of frames from the input video, extraction of per-frame features by the shared CNN, temporal analysis of the feature sequence, and classification of the video as real or fake. The explicit gating in Eqs. (4) and (5) allows the recurrent heads to retain information about inconsistencies that appear intermittently across frames, which the order-invariant average in Eq. (6) cannot represent.

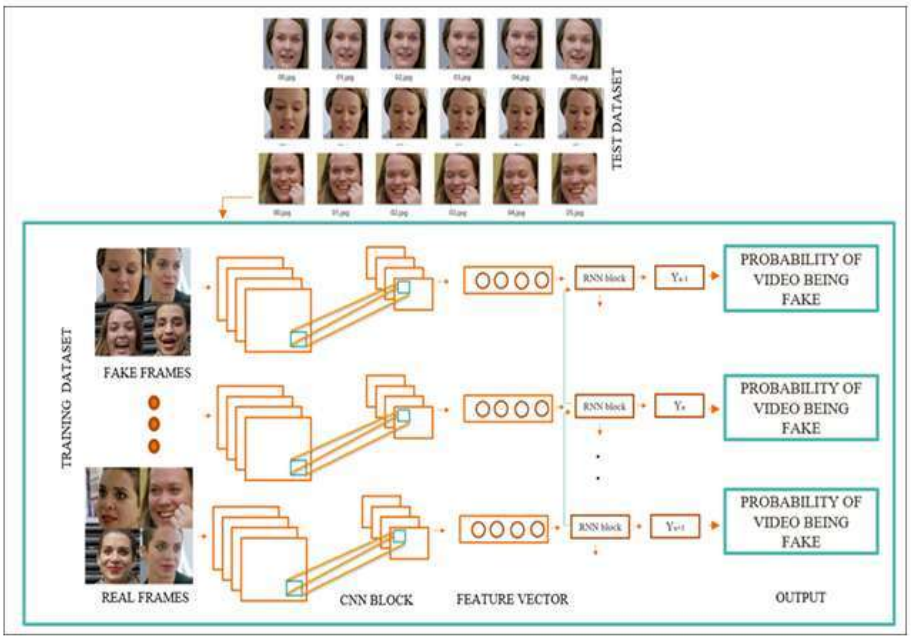


Fig. 3. Proposed convolutional–recurrent detection architecture. Face crops from consecutive frames are encoded by a shared XceptionNet block into feature vectors, which are analysed by a recurrent (LSTM or GRU) block to estimate the probability that the video is fake.

3.6 Training protocol, optimiser selection and multi-domain experiment

All sequence models were initialised with ImageNet weights and trained with the binary cross-entropy loss of Eq. (2). Early stopping on the validation split was employed to prevent overfitting, and test performance was recorded after 30, 50 and 70 epochs. To select the optimiser, each architecture was trained on DFD for 70 epochs with eight optimisers—stochastic gradient descent (SGD), RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam and FTRL—and compared in terms of test accuracy and log-loss; the best optimiser was used in all subsequent experiments. In Experiment C, the CNN-LSTM model was trained, with and without transfer learning, on Celeb-DF, DFDC, DFD and FF++ individually and on a pooled custom set that combines samples from all four, and each model was tested on every benchmark test split. The pooled model was additionally tested on an unseen set ("Custom + New Samples") that combines blended benchmark samples with fake videos collected from YouTube.

3.7 Evaluation metrics and implementation

Performance was assessed using accuracy, precision (TP/(TP + FP)), recall (TP/(TP + FN)), F1-score (the harmonic mean of precision and recall) and log-loss, where TP, FP and FN denote true positives, false positives and false negatives with respect to the fake class. Confusion matrices, receiver operating characteristic (ROC) curves and precision–recall (PR) curves were used for further analysis. Log-loss penalises confident incorrect predictions and is therefore complementary to accuracy. The models were implemented in Python using TensorFlow/Keras, with Dlib and OpenCV

for face detection and image processing, and were trained on an NVIDIA DGX-1 system (eight Tesla V100 GPUs), with additional experiments on an IBM Power AC922 server and Google Colaboratory.

4. RESULTS AND DISCUSSION

4.1 Effect of transfer learning on convergence

Fig. 4 compares the validation accuracy of the single-frame XceptionNet after one training epoch with and without transfer learning. For all three benchmarks, the ImageNet-initialised model starts from a markedly higher accuracy, with a margin of approximately 9–13 percentage points as read from the bar chart. This behaviour is consistent with the activations of the first convolutional layers of the pre-trained network, which already respond to edges, textures and facial contours; freezing these layers during the first stage of training allows the classification head to exploit generic features immediately, whereas a randomly initialised network must first learn low-level filters from the deepfake data. From an engineering standpoint, faster convergence reduces training cost on large frame corpora (Table 1) and makes it practical to retrain detectors as new generation methods appear.

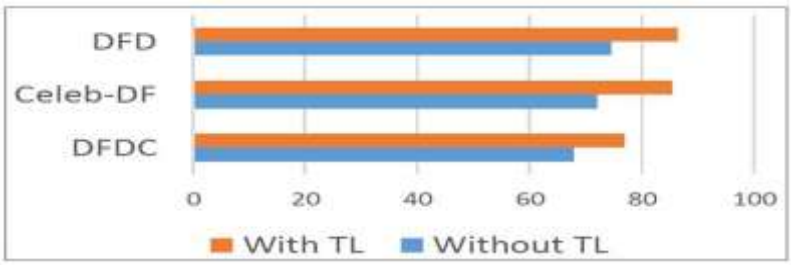


Fig. 4. Validation accuracy (%) of the single-frame XceptionNet after one epoch with and without transfer learning (TL) on DFD, Celeb-DF and DFDC-Preview.

4.2 Self- and cross-dataset performance of the single-frame detector

Table 3 reports the accuracies of all trained models on all test sets. Three observations stand out. First, transfer learning improves accuracy in 18 of the 20 train–test configurations, with a mean gain of 4.97 percentage points. The largest self-test improvement is obtained for Celeb-DF, where accuracy increases from 83.49% to 94.60% (+11.11 percentage points, or a relative improvement of (94.60 − 83.49)/83.49 = 13.3%). The self-test gains for DFDC-Preview and DFD are +6.57 and +1.53 percentage points, respectively; the small gain for DFD reflects the fact that the model trained from scratch already exceeds 94% on this dataset. The two exceptions are the DFDC→Celeb-DF (65.53% → 64.35%) and Celeb-DF→DFDC (68.44% → 65.84%) pairs, which involve the two distributions with the highest manipulation quality; in these cases fine-tuning may have strengthened source-specific cues that do not transfer between the two synthesis procedures.

Table 3. Classification accuracy (%) of the single-frame XceptionNet without (w/o) and with transfer learning (TL) for self- and cross-dataset testing. Rows denote training distributions and columns denote test sets.

Training set	DFDC-P		Celeb-DF		DFD		Custom		Combined	
	w/o TL	TL	w/o TL	TL	w/o TL	TL	w/o TL	TL	w/o TL	TL
DFDC-P	78.13	84.70	65.53	64.35	59.60	64.14	60.02	65.31	66.11	69.81
Celeb-DF	68.44	65.84	83.49	94.60	45.76	53.41	53.05	58.32	63.54	68.91
DFD	58.24	66.23	62.36	65.33	94.33	95.86	67.86	70.57	70.16	74.03
Combined	79.86	82.47	85.72	94.62	88.52	94.66	63.19	73.20	79.62	86.49

Second, the cross-dataset results expose severe dataset shift. With transfer learning, the mean cross-test accuracy of the single-source models on the two other benchmarks is 64.25% for DFDC-Preview, 59.62% for Celeb-DF and 65.78% for DFD, which corresponds to reductions of 20.45, 34.97 and 30.08 percentage points relative to their self-test accuracies. The weakest pair is Celeb-DF→DFD (45.76% without and 53.41% with transfer learning), i.e. close to chance level. As shown in Fig. 1, these two datasets occupy opposite quadrants: Celeb-DF combines high manipulation quality with lower visual quality, whereas DFD offers high visual quality but less refined manipulations. A detector trained on Celeb-DF therefore learns to rely on subtle blending cues within degraded frames, which are absent from the sharp but more obviously manipulated DFD faces. Conversely, the DFD model provides the best cross-test result

(66.23% on DFDC-Preview) and the highest custom-set and combined-set accuracies among the single-source models (70.57% and 74.03%), suggesting that training on high-visual-quality footage from a moderately large pool of actors yields features that transfer comparatively well. The relatively low self-test accuracy of the DFDC-Preview model (84.70%) is attributable to the high manipulation quality of this dataset and to the frame-rate, resolution and encoding degradations applied to its test clips.

Third, increasing the diversity of the training distribution is the most effective measure for generalisation. The model trained on the combined distribution attains 94.62% on Celeb-DF and 94.66% on DFD—essentially equal to the corresponding single-source self-test accuracies—and 86.49% on the combined test set, which is 12.46 percentage points higher than the best single-source model (74.03%). On the custom test set, which contains no benchmark data, the combined model with transfer learning reaches 73.20%, an improvement of 10.01 percentage points (15.8% relative) over its counterpart without transfer learning (63.19%) and of 2.63 percentage points over the best single-source model. The combination of source diversity and pre-trained initialisation is therefore complementary: diversity exposes the network to multiple artefact types, while ImageNet initialisation acts as a regulariser that limits overfitting to any single generation pipeline. As a consistency check, applying Eq. (3) to the self- and cross-set accuracies in Table 3 with the test-set sizes of Table 1 reproduces all reported combined-set accuracies with transfer learning (e.g. 74.03% for DFD and 86.49% for the combined model). Nevertheless, even the best model loses about 13 percentage points between the combined and custom test sets (86.49% versus 73.20%), which indicates that in-the-wild and self-generated deepfakes remain substantially harder than benchmark samples.

4.3 Optimiser selection for sequence classifiers

Table 4 compares the eight optimisers after 70 epochs of training on DFD. Nadam yields the highest accuracy for all three architectures (82.03% for CNN, 85.20% for CNN-LSTM and 83.59% for CNN-GRU) and the lowest log-loss for CNN-LSTM (0.493) and CNN-GRU (0.466); for the CNN baseline, its log-loss (0.512) is second only to RMSprop (0.506). Relative to Adam, Nadam improves accuracy by 4.69, 3.95 and 9.38 percentage points and reduces log-loss by 3.9%, 14.3% and 8.1% for CNN, CNN-LSTM and CNN-GRU, respectively. Adadelta and Adagrad perform poorly, with CNN-GRU accuracy falling to 41.41% for both; within a fixed budget of 70 epochs, their monotonically shrinking effective learning rates appear to stall training before the recurrent weights converge. SGD without adaptive scaling is similarly ineffective for the CNN baseline (50.78%).

Table 4. Test accuracy (%) and log-loss of the three architectures on DFD after 70 epochs for eight optimisers. The best value in each column is shown in bold.

Optimiser	CNN		CNN-LSTM		CNN-GRU	
	Accuracy	Log-loss	Accuracy	Log-loss	Accuracy	Log-loss
SGD	50.78	0.723	72.65	0.600	60.15	0.665
RMSprop	81.25	**0.506**	81.25	0.534	78.91	0.506
Adam	77.34	0.533	81.25	0.575	74.21	0.507
Adadelta	50.00	0.680	56.25	0.684	41.41	0.828
Adagrad	57.03	0.700	59.38	0.686	41.41	0.714
Adamax	75.78	0.587	75.78	0.546	77.34	0.563
Nadam	**82.03**	0.512	**85.20**	**0.493**	**83.59**	**0.466**
FTRL	75.80	0.657	77.34	0.657	71.88	0.634

The sensitivity to the optimiser differs between architectures: the spread between the best and worst optimiser is 28.95 percentage points for CNN-LSTM, 32.03 for CNN and 42.18 for CNN-GRU. The LSTM model is thus the most robust to optimiser choice, and it is the only architecture for which plain SGD reaches a usable accuracy (72.65%). The advantage of Nadam can be explained by its combination of adaptive per-parameter step sizes with Nesterov momentum, whose look-ahead gradient dampens oscillations caused by noisy mini-batch gradients; such noise is to be expected when batches mix faces from different actors, illumination conditions and manipulation methods. Nadam was therefore used for all subsequent sequence experiments.

4.4 Comparison of CNN, CNN-LSTM and CNN-GRU

Table 5 summarises the accuracy of the three architectures at 30, 50 and 70 epochs on the four datasets, together with precision, recall and F1-score at 70 epochs, and Fig. 5 visualises the accuracy trends. Accuracy is non-decreasing with the training budget in all twelve model–dataset combinations, confirming that none of the models had saturated at 30 epochs. The largest budget-related gains occur for the more difficult datasets, e.g. +7.41 percentage points for CNN on FF++ and +6.3 percentage points for CNN-LSTM on DFDC between 30 and 70 epochs, whereas the CNN-LSTM on FF++ already reaches 68.8% at 30 epochs and improves by only 0.4 percentage points thereafter.

Table 5. Test accuracy (%) of the three architectures at 30, 50 and 70 epochs (Nadam optimiser) and precision, recall and F1-score at 70 epochs. The best 70-epoch accuracy for each dataset is shown in bold.

Dataset	Model	Acc. 30 ep.	Acc. 50 ep.	Acc. 70 ep.	Precision	Recall	F1-score
DFD	CNN	80.50	81.20	82.00	0.887	0.734	0.803
	CNN-LSTM	81.25	82.80	**85.20**	0.895	0.797	0.843
	CNN-GRU	80.50	81.20	83.60	0.830	0.688	0.752
DFDC	CNN	57.80	58.60	61.70	—[a]	—[a]	—[a]
	CNN-LSTM	57.80	58.60	**64.10**	—[a]	—[a]	—[a]
	CNN-GRU	56.20	57.20	61.50	0.613	0.594	0.603
FF++	CNN	58.59	59.00	66.00	0.610	0.891	0.724
	CNN-LSTM	68.80	69.20	**69.20**	0.672	0.750	0.709
	CNN-GRU	52.00	55.90	58.20	0.574	0.609	0.591
Custom	CNN	69.10	70.60	72.10	0.621	0.591	0.606
	CNN-LSTM	69.90	71.30	**73.20**	0.653	0.688	0.670
	CNN-GRU	68.80	69.90	69.90	0.604	0.644	0.623

[a] DFDC accuracies for CNN and CNN-LSTM are taken from the epoch-wise accuracy analysis (Fig. 5); the corresponding precision/recall values are not reported because the source tabulation for these two models could not be unambiguously attributed.

At 70 epochs, CNN-LSTM attains the highest accuracy on every dataset: 85.2% on DFD, 64.1% on DFDC, 69.2% on FF++ and 73.2% on the custom dataset. Its margins over the frame-level CNN are 3.2, 2.4, 3.2 and 1.1 percentage points, respectively, and its mean accuracy across the four datasets (72.92%) exceeds those of CNN (70.45%) and CNN-GRU (68.30%). In contrast, CNN-GRU outperforms the CNN baseline only on DFD (+1.6 percentage points), is practically equal on DFDC (−0.2), and is clearly inferior on FF++ (−7.8) and on the custom dataset (−2.2). The benefit of recurrent modelling is therefore not automatic but depends on the capacity of the recurrent unit. A plausible explanation is that the separate cell state and output gate of the LSTM (Eq. (4)) allow evidence of intermittent artefacts—flicker, landmark jitter or colour inconsistency appearing in only a few frames—to be retained across the sequence, whereas the coupled update of the GRU (Eq. (5)) interpolates between old and new states and may overwrite such sparse evidence. The poor GRU result on FF++, which contains four manipulation methods including expression-level reenactments (Face2Face, NeuralTextures) whose temporal signatures are subtle, is consistent with this interpretation. Dataset difficulty strongly influences all architectures. All three models exceed 80% on DFD, whereas accuracies on DFDC and FF++ remain between 58% and 69%. The high visual and manipulation quality of DFDC, its diversity of actors, lighting conditions and backgrounds, and the heterogeneous compression of FF++ make the manipulations considerably harder to detect with 128 × 128 face crops. The custom dataset, which mixes all benchmark sources, yields intermediate accuracies (69.9–73.2%), in line with the observation from Experiment A that mixed distributions are harder to fit than a single distribution but provide better generalisation.

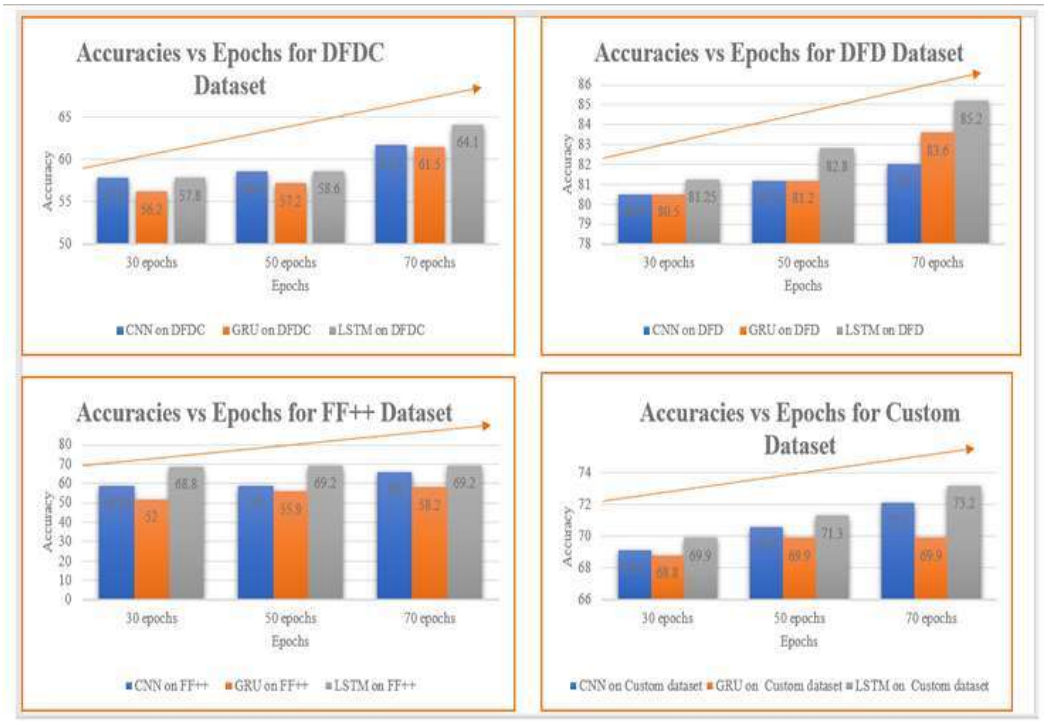


Fig. 5. Test accuracy versus training epochs for CNN, CNN-GRU and CNN-LSTM on the DFDC, DFD, FF++ and custom datasets.

The precision–recall balance also differs between datasets. On DFD, all models have higher precision than recall (e.g. 0.895 versus 0.797 for CNN-LSTM), meaning that videos flagged as fake are usually fake but a proportion of fakes is missed. On FF++, the CNN baseline exhibits the opposite behaviour (precision 0.610, recall 0.891), flagging most fakes at the cost of many false alarms, whereas CNN-LSTM achieves a more balanced trade-off (0.672 and 0.750). In practical terms, a content-moderation system would favour high recall so that suspicious videos are forwarded for human verification, while a forensic application would favour high precision; the decision threshold of Eq. (6) can be tuned accordingly.

4.5 Error analysis and training behaviour

Fig. 6 compares the confusion matrices of the CNN and CNN-LSTM models on a balanced set of 128 DFD test samples (64 real and 64 fake) at 70 epochs. Both models classify authentic videos identically (58 true negatives and 6 false positives, i.e. a false-positive rate of 9.4%). The improvement of CNN-LSTM is confined to the fake class: false negatives decrease from 17 to 13, a reduction of $(17 - 13)/17 = 23.5\%$, and true positives increase from 47 to 51. Consequently, recall rises from 0.734 to 0.797 (+8.6% relative), precision from 0.887 to 0.895, and the F1-score from 0.803 to 0.843 (+5.0% relative). This pattern supports the interpretation that temporal modelling primarily helps to detect fakes whose artefacts are not evident in individual frames but emerge across the sequence, while it does not alter the treatment of authentic videos.

The training behaviour of CNN-LSTM on DFD is shown in Fig. 7. The training loss decreases steadily from approximately 0.70 to approximately 0.45 over 70 epochs, while the training accuracy increases from about 0.55 to above 0.80, with most of the improvement occurring in the first 20–30 epochs. The validation curves follow the same trend but oscillate around the training curves, and the validation loss stabilises at a slightly higher level than the training loss after about 30 epochs. The oscillations are attributable to the small validation split (140 videos for DFD, Table 2), in which each misclassified video changes accuracy by about 0.7 percentage points, and the modest gap between training and validation loss indicates limited overfitting, which supports the use of dropout and early stopping.

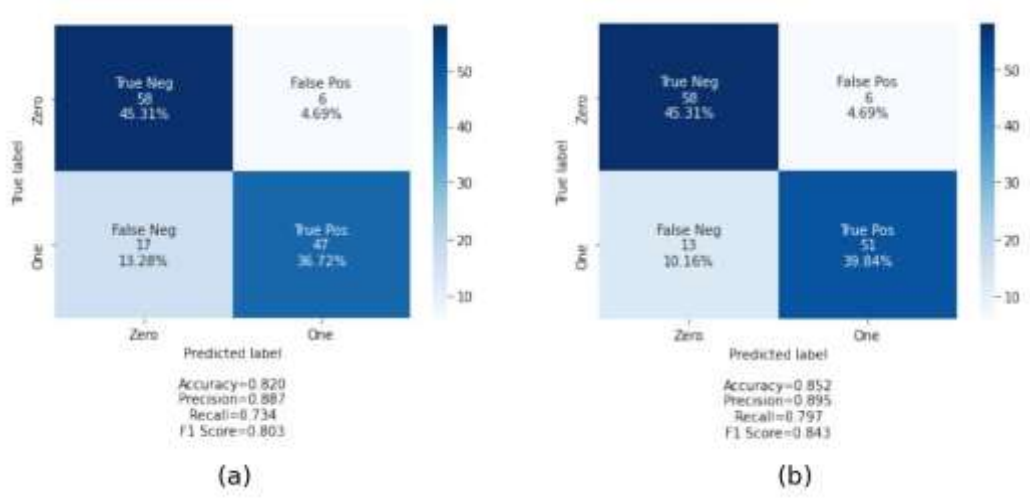


Fig. 6. Confusion matrices on the DFD test samples after 70 epochs: (a) frame-level CNN; (b) CNN-LSTM. Label "One" denotes the fake class.

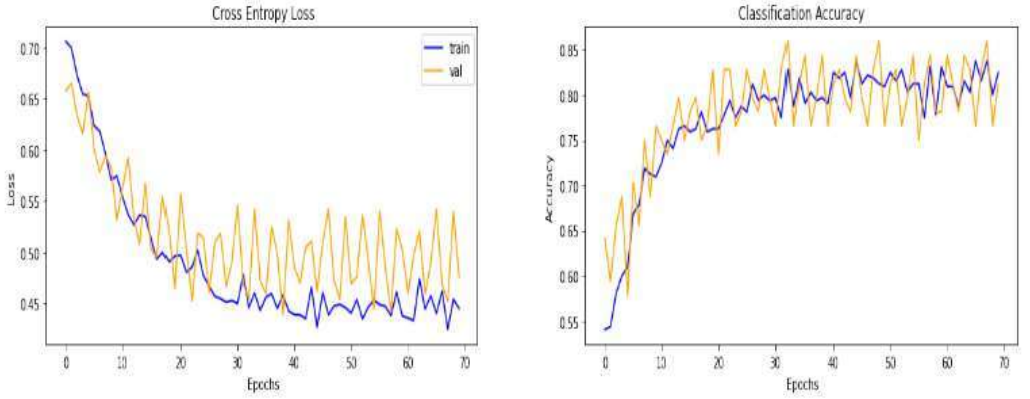


Fig. 7. Training (blue) and validation (orange) cross-entropy loss and classification accuracy of the CNN-LSTM model on DFD over 70 epochs.

The ROC and PR curves of CNN-LSTM are presented in Fig. 8. The ROC curve rises steeply at low false-positive rates and bows towards the upper-left corner, indicating good separability of the two classes. The PR curve shows that precision remains close to or above 0.9 up to a recall of roughly 0.5 and then decreases gradually, falling more steeply beyond a recall of about 0.85. For deployment, the PR curve provides a direct basis for selecting the decision threshold according to the relative costs of missed fakes and false alarms in the target application.

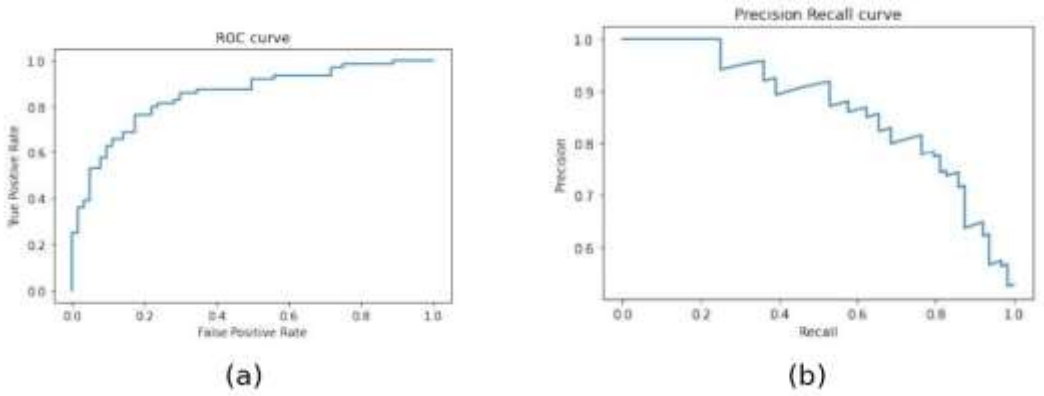


Fig. 8. Performance curves of the CNN-LSTM model on DFD: (a) ROC curve; (b) precision–recall curve.

4.6 Multi-domain generalisation of the CNN-LSTM detector

Table 6 reports the results of Experiment C. When the CNN-LSTM is trained on a single benchmark, its accuracy on the other benchmarks remains between 58.86% and 67.07% even with transfer learning (and falls to 52.45% without it), reproducing the dataset-shift effect observed for the single-frame detector in Table 3. The DFD-trained model again

generalises best among the single-source models (69.58% on the custom set with transfer learning), in agreement with Experiment A, and its self-test accuracy (85.20%) coincides with the DFD result of Table 5. Transfer learning improves accuracy in all 25 train–test configurations of Table 6, with a mean gain of 3.19 percentage points; the largest gains occur for cross-dataset pairs such as DFD→DFDC (+7.79 percentage points) and Celeb-DF→DFD (+6.41 percentage points), confirming that pre-trained initialisation is most valuable precisely where domain shift is largest.

Table 6. Accuracy (%) of the CNN-LSTM detector trained on individual and pooled datasets, without (w/o) and with transfer learning (TL), for multi-domain testing. "Custom" denotes the pooled DFD + DFDC + Celeb-DF + FF++ data; "Custom + New" adds unseen blended and in-the-wild (YouTube) samples.

Training data	Celeb-DF		DFDC		DFD		FF++		Custom		Custom + New	
	w/o	TL	w/o	TL	w/o	TL	w/o	TL	w/o	TL	w/o	TL
Celeb-DF	63.06	64.56	57.77	63.37	52.45	58.86	59.75	61.45	61.00	62.38	–	–
DFDC	62.00	63.45	62.30	64.10	61.00	62.67	58.89	60.78	58.68	61.24	–	–
DFD	60.77	63.98	57.65	65.44	80.35	85.20	65.90	67.07	66.04	69.58	–	–
FF++	62.36	65.89	60.76	62.87	63.98	65.57	65.05	69.20	61.68	65.45	–	–
Custom (pooled)	80.65	82.45	79.64	83.87	80.56	86.98	83.84	86.56	69.78	72.59	78.67	**81.54**

Training on the pooled data changes the picture substantially. With transfer learning, the pooled model achieves between 82.45% (Celeb-DF) and 86.98% (DFD) on the individual benchmark test sets, i.e. approximately 17–28 percentage points above the corresponding cross-dataset results of the single-source models. On the pooled custom test set, it reaches 72.59%, which is 3.01 percentage points above the best single-source model and 7.93 percentage points above the mean of the four single-source models (64.66%). Most importantly, on the unseen "Custom + New Samples" set, which includes in-the-wild deepfakes from YouTube, the pooled CNN-LSTM with transfer learning attains 81.54%, an improvement of 2.87 percentage points (3.6% relative) over the same model without transfer learning (78.67%). Taken together with Table 3, these results show that the diversity of the training sources is the dominant factor in generalisation, transfer learning provides a consistent additional gain, and temporal modelling contributes a further, smaller but consistent improvement over frame-level classification.

4.7 Comparison with reported approaches and limitations

Several studies report higher accuracies than those obtained here, for example 97.1% for a Conv-LSTM on a custom dataset derived from HOHA and online videos [10], 91.88% for an MTCNN–CNN–GRU pipeline with automatic face weighting on DFDC [32], and values above 90% for landmark-based Conv-LSTM variants on DFDC [22]. These figures were obtained under intra-dataset protocols on specific subsets, whereas the present evaluation emphasises cross-dataset and in-the-wild testing, in which the same class of detectors loses 20–35 percentage points (Table 3). Direct numerical comparison is therefore not meaningful; the present results should instead be interpreted as evidence of how much of the reported performance is retained when the test distribution changes, and of which design choices mitigate the loss.

The study has limitations that should be considered when interpreting the results. The video-level test sets of Experiment B are modest in size (140 videos for DFD and DFDC), so differences of one to two percentage points between architectures correspond to only a few videos and were not tested for statistical significance over repeated runs. A single backbone (XceptionNet) and a single input resolution (128 × 128 pixels) were used, and the sequence length and recurrent-layer size were not optimised. Robustness to specific post-processing operations, such as compression level or resolution changes, was not isolated, and audio cues were not exploited. Finally, the custom test sets, although deliberately heterogeneous, are small relative to the benchmark corpora.

5. CONCLUSIONS

The present study investigated the generalisability of video deepfake detection using a transfer-learned XceptionNet backbone and convolutional–recurrent extensions evaluated under self-, cross- and multi-domain protocols. The main conclusions are as follows:

(1) Dataset shift is the dominant limitation of learning-based deepfake detectors. Single-source single-frame models lost 20.45–34.97 percentage points when tested on other benchmarks, and the Celeb-DF→DFD pair fell to near-

chance accuracy (53.41% with transfer learning), reflecting the opposite visual- and manipulation-quality characteristics of these datasets.

(2) ImageNet transfer learning with staged fine-tuning accelerated convergence and improved accuracy in 18 of 20 single-frame configurations (mean +4.97 percentage points) and in all 25 multi-domain CNN-LSTM configurations (mean +3.19 percentage points), with the largest gains in cross-dataset settings.

(3) Training-source diversity was the most effective measure for generalisation: the combined-distribution model reached 86.49% on the combined test set and 73.20% on the custom test set, and the pooled CNN-LSTM reached 81.54% on unseen blended and in-the-wild samples.

(4) Nadam was the most effective of eight optimisers for all three architectures, whereas Adadelta and Adagrad failed to converge adequately within 70 epochs.

(5) CNN-LSTM achieved the highest accuracy on all four datasets (85.2% on DFD, with precision 0.895, recall 0.797 and F1-score 0.843), reducing false negatives on DFD by 23.5% relative to the frame-level CNN, whereas CNN-GRU did not consistently outperform the baseline.

SCOPE FOR FUTURE WORK: Future work will address larger and more diverse in-the-wild test sets, repeated-run statistical analysis, alternative backbones such as EfficientNet, data augmentation targeted at generator-specific artefacts, videos containing multiple faces, temporal convolutional networks, and joint audio–visual detection.

Data availability: The DFD, Celeb-DF, DFDC and FaceForensics++ datasets are publicly available from their respective providers

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. 27th Int. Conf. Neural Information Processing Systems (NIPS), vol. 2, 2014, pp. 2672–2680.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4396–4405.

[3] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2387–2395.

[4] "faceswap-GAN," GitHub repository. [Online]. Available: https://github.com/shaoanlu/faceswap-GAN/

[5] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi, "Deep learning for deepfakes creation and detection," arXiv preprint arXiv:1909.11573, 2019.

[6] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in Proc. IEEE Winter Applications of Computer Vision Workshops (WACVW), 2019, pp. 83–92, doi: 10.1109/WACVW.2019.00020.

[7] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in Proc. IEEE Int. Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7.

[8] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807.

[9] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2019, pp. 1–11.

[10] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in Proc. 15th IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639163.

[11] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2019, pp. 80–87.

[12] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," IEEE Trans. Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3009287.

[13] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A new dataset for DeepFake forensics," arXiv preprint arXiv:1909.12962, 2019.

[14] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The Deepfake Detection Challenge (DFDC) preview dataset," arXiv preprint arXiv:1910.08854, 2019.

[15] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," Pattern Recognition, vol. 45, no. 1, pp. 521–530, 2012.

[16] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," Machine Learning, vol. 79, no. 1–2, pp. 151–175, 2010.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[18] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[19] J. Donahue et al., "Long-term recurrent convolutional networks for visual recognition and description," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 677–691, 2017, doi: 10.1109/TPAMI.2016.2599174.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[22] M. F. Hashmi, B. K. K. Ashish, A. G. Keskar, N. D. Bokde, J. H. Yoon, and Z. W. Geem, "An exploratory analysis on visual counterfeits using Conv-LSTM hybrid architecture," IEEE Access, vol. 8, pp. 101293–101308, 2020, doi: 10.1109/ACCESS.2020.2998330.

[23] Ondyari, "FaceForensics (including the Google/Jigsaw DeepFake Detection dataset)," GitHub repository. [Online]. Available: https://github.com/ondyari/FaceForensics/ (accessed May 10, 2021).

[24] "Deepfake Detection Challenge dataset," Facebook AI. [Online]. Available: https://ai.facebook.com/datasets/dfdc/ (accessed May 10, 2021).

[25] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," IEEE Trans. Information Forensics and Security, vol. 5, no. 3, pp. 480–491, 2010.

[26] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in Proc. 4th ACM Workshop on Information Hiding and Multimedia Security, 2016, pp. 5–10.

[27] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 8261–8265, doi: 10.1109/ICASSP.2019.8683164.

[28] D. Pan, L. Sun, R. Wang, X. Zhang, and R. O. Sinnott, "Deepfake detection through deep learning," in Proc. IEEE/ACM Int. Conf. Big Data Computing, Applications and Technologies (BDCAT), 2020, pp. 134–143.

[29] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," arXiv preprint arXiv:1905.11946, 2019.

[30] J. H. Bappy, C. Simons, L. Nataraj, B. S. Manjunath, and A. K. Roy-Chowdhury, "Hybrid LSTM and encoder–decoder architecture for detection of image forgeries," IEEE Trans. Image Processing, vol. 28, no. 7, pp. 3286–3300, 2019.

[31] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in Proc. IEEE Int. Workshop on Information Forensics and Security (WIFS), 2016, pp. 1–6, doi: 10.1109/WIFS.2016.7823911.

[32] D. M. Montserrat et al., "Deepfakes detection with automatic face weighting," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2851–2859, doi: 10.1109/CVPRW50498.2020.00342.

[33] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking," arXiv preprint arXiv:1806.02877, 2018.

[34] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based CNN," in Proc. IEEE/CVF Int. Conf. Computer Vision Workshops (ICCVW), 2019.

[35] S. Das, A. Datta, M. S. Islam, and M. R. Amin, "Improving DeepFake detection using dynamic face augmentation," 2021.

[36] S. Suratkar, E. Johnson, K. Variyambat, M. Panchal, and F. Kazi, "Employing transfer-learning based CNN architectures to enhance the generalizability of deepfake detection," in Proc. 11th IEEE Int. Conf. Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1–9, doi: 10.1109/ICCCNT49239.2020.9225400.

[37] S. Suratkar and F. Kazi, "Deep fake video detection using transfer learning approach," Arabian Journal for Science and Engineering, 2022, doi: 10.1007/s13369-022-07321-3.

[38] S. Suratkar, S. Bhiungade, J. Pitale, K. Soni, T. Badgujar, and F. Kazi, "Deep-fake video detection approaches using convolutional–recurrent neural networks," Journal of Control and Decision, 2022, doi: 10.1080/23307706.2022.2033644.

[39] D. E. King, "Dlib-ml: A machine learning toolkit," Journal of Machine Learning Research, vol. 10, pp. 1755–1758, 2009.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2015, pp. 1026–1034.