Limitations of Transfer Learning for Tabular Insurance Fraud Detection Using Adapted CNNs

Swayam Chaurasia
22BCE8346

Sonali Singh
22BCE8614

Mukul Shekhar
22BCE7270

***Abstract*—This paper presents an investigation into the application of Transfer Learning (TL) for insurance fraud detection by adapting two image classification architectures, GoogleNet (InceptionV3) and a VGG19-based TableNet-Inspired model, to tabular insurance claims data. Tabular features were padded to 64 dimensions and reshaped into 8×8 grids, then spatially resized to match CNN input requirements (299×299×3 and 224×224×3, respectively). We compare both adapted CNNs, initialized with ImageNet weights, against a high-performing custom Deep Neural Network (DNN). The models are trained on a dataset augmented using the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. Evaluation reveals that the tabular-to-image conversion performed poorly in our experiments, yielding near-chance accuracy (Accuracy ≈ 0.50–0.52). In stark contrast, the custom DNN achieves superior performance (Recall = 0.9370, F1-Score = 0.9592), highlighting the severe limitations of such non-contextual spatial transformations in adapting vision models to tabular domains.**

***Index Terms*—Fraud Detection, Transfer Learning, GoogleNet, TableNet, Tabular-to-Image Conversion, Deep Learning, Custom DNN**

I. Introduction

Insurance fraud represents a major financial challenge globally. While traditional machine learning (ML) excels at tabular data, this study explores the potential of Transfer Learning (TL) by leveraging highly optimized architectures from the image domain, such as GoogleNet (InceptionV3) and VGG19 (as the feature backbone for a TableNet-inspired model). This approach aims to utilize the powerful, hierarchical feature extraction capabilities of Convolutional Neural Networks (CNNs).

The core challenge lies in effectively transforming multi-dimensional tabular data into a dense, channel-rich format suitable for image-based CNNs. This paper details a systematic methodology for this conversion and critically evaluates the efficacy of two adapted CNN architectures in classifying fraudulent insurance claims by comparing their results against a dedicated, custom Deep Neural Network (DNN).

While traditional machine learning (ML) excels at tabular data, this study investigates whether transfer learning from image-based architectures (InceptionV3, VGG19) can model fraud patterns in tabular insurance claims. We hypothesize that spatial transformation might partially preserve inter-feature interactions; however, the lack of inherent spatial relationships poses a fundamental limitation. This paper empirically evaluates these adapted CNNs against a purpose-built DNN baseline to quantify the transfer learning gap.

II. Related Work

Adapting computer vision models for non-image data has recently gained attention. High-performing architectures like InceptionV3 and VGG19 possess deep layers capable of learning complex data hierarchies. Adapting tabular data for these CNNs typically involves mapping features onto a 2D or 3D grid, treating the resulting structure as an image. While this approach allows the use of established CNN architectures, its success depends on how well the feature-to-pixel mapping preserves underlying data relationships.

Our work employs a feature-padding and resizing technique, using standard Transfer Learning (freezing the base ImageNet-initialized weights) to rigorously test the viability of these adapted architectures.

III. Methodology

A. Dataset and Initial Preprocessing

The dataset used consists of anonymized insurance claim records with 38 features and a binary target variable, *fraud_reported*. The original dataset contained 1000 records. Through the augmentation process (detailed in Section III-B), the final working dataset was expanded to 11,542 samples.

Missing Value Management: Missing values were initially marked with a null indicator. Numerical features were imputed using the mode of each feature, while categorical features were imputed with a Boolean *False* value.

Encoding: Target Encoding was applied to categorical features to preserve ordinal relationships while maintaining consistency across categorical levels.

Feature Engineering and Scaling: Highly correlated features were removed to reduce multicollinearity. All numerical features were standardized using *StandardScaler* to ensure uniform feature distributions across the dataset.

B. Data Augmentation

To address both the small size of the original dataset (1000 records) and the inherent class imbalance, a multi-stage augmentation strategy was employed.
The process proceeded in two sequential stages:

1) **Initial Augmentation and Data Preparation (Resulting in Intermediate Dataset)**
   - **Preprocessing:** Initial data cleanup, including the imputation of missing values (using mode for numerical features and Boolean `False` for categorical

features) and Target Encoding of categorical features, was performed to the original dataset resulting in 35 features plus the target column.
  - **SMOTE Application:** The pre-processed dataset was initially augmented using the Synthetic Minority Over-sampling Technique (SMOTE) to balance the classes.
  - **First Expansion:** This stage resulted in an intermediate dataset (5518 records) that was used as the base for the subsequent steps.

2) **Final Expansion and Noise Injection**
  - **Gaussian Noise Injection:** Gaussian noise (scaled to 5% of the standard deviation) was added to the numerical features of the intermediate dataset to improve model robustness and simulate real-world variability.
  - **Repeated SMOTE Application:** The SMOTE technique was applied again to the minority class of the noise-injected intermediate dataset. This secondary application of SMOTE further increased the dataset size and reinforced the class balance necessary for stable deep learning convergence.
  - **Final Dataset:** The final dataset (11,542 samples) was obtained by combining the noise-injected data with the synthetic samples generated from the second SMOTE application. This methodology ensures the final dataset is large enough and sufficiently balanced for the comparative deep learning study.

C. Tabular-to-Image Conversion Pipeline (CNNs)

The consistent feature set was padded to 64 (an 8×8 grid) and reshaped to 8×8×1. It was then converted to RGB (8×8×3) and spatially resized for each CNN. ImageNet-specific preprocessing was applied to the resulting images.

The 29 artificial features required to complete the 8×8 grid were created using zero-padding (constant value=0). This method was chosen for its simplicity, but it is a primary source of the "non-contextual spatial transformation" that corrupts the feature space

TABLE I
CNN INPUT SPECIFICATIONS AND PREPROCESSING FUNCTIONS

Model	Target Input Size	Base Model	Preprocessing Function
GoogleNet	299×299×3	InceptionV3	`inception_v3.preprocess_input`
TableNet	224×224×3	VGG19	`vgg19.preprocess_input`

D. Model Architectures

Three primary deep learning models were used for comparison, maintaining architectural consistency in their classification heads where applicable:

- **Adapted GoogleNet (InceptionV3) – Transfer Learning:** The InceptionV3 model was loaded with an input shape of (299, 299, 3) and pre-trained ImageNet weights. The base model was frozen during initial training (Phase 1 TL) for stability. A custom classification head—designed to process the deep features—was attached, consisting of: **Architecture:** GlobalAveragePooling2D → Dense(128, ReLU) → Dense(64, ReLU) → Dense(1, Sigmoid)
- **TableNet-Inspired Model (VGG19):** This model used VGG19 as its frozen ImageNet-initialized feature extractor, compatible with a (224, 224, 3) input. A custom classification head identical to GoogleNet's was attached: **Architecture:** GlobalAveragePooling2D → Dense(128, ReLU) → Dense(64, ReLU) → Dense(1, Sigmoid)
- **Custom Deep Neural Network (DNN):** A dedicated tabular baseline with the following architecture: **Architecture:** Dense(64, ReLU) → Dense(32, ReLU) → Dense(1, Sigmoid)

E. Training and Evaluation

Both CNNs used a memory-safe batch generator, while the DNN operated directly on numerical arrays. All models were trained for 15 epochs using the Adam optimizer and binary cross-entropy loss. Evaluation metrics included Accuracy, Precision, Recall, and F1-Score, focusing on the fraud (positive) class.

IV. RESULTS AND DISCUSSION (REVISED)

TABLE II
PERFORMANCE METRICS OF ADAPTED CNNS VS. CUSTOM DNN

Model	Accuracy	Precision (C1)	Recall (C1)	F1-Score (C1)
Custom DNN	0.9599	0.9826	0.9370	0.9592
TableNet (VGG19)	0.5201	0.5143	0.8579	0.6430
GoogleNet (InceptionV3)	0.5004	0.5033	0.6521	0.5681

The results demonstrate an overwhelming performance gap between the dedicated tabular model and the adapted CNNs.

Custom DNN Success: The DNN achieved excellent performance (F1-Score = 0.9592), validating both the feature quality and the network's efficiency. Despite only 15 epochs, it converged rapidly, highlighting its strong suitability for tabular data.

CNN Failure Analysis (Both Models): Both GoogleNet and the TableNet-inspired VGG19 models performed near-chance (Accuracy ≈ 0.50–0.52).

TableNet (VGG19) Failure: While it achieved a high Recall (0.8579), indicating sensitivity to fraud cases, its low Precision (0.5143) shows nearly half of the fraud predictions were false positives. The resulting F1-Score (0.6430) reflects ineffective generalization, leaning toward a positive class bias.

GoogleNet (InceptionV3) Failure: This model performed closest to random chance (Accuracy = 0.5004, F1-Score = 0.5681), confirming that tabular-to-image conversion destroyed meaningful feature representation.

The consistent underperformance of both CNNs—even with ImageNet TL—proves that non-contextual spatial transformation severely corrupts the feature space, rendering convolutional backbones ineffective for tabular data.

V. Conclusion

This study found that adapting deep image models (GoogleNet and TableNet-Inspired VGG19) via tabular-to-image conversion for insurance fraud detection fails, achieving only near-random accuracy. In contrast, the custom DNN achieved vastly superior performance (F1-Score = 0.9592). This strongly reinforces that non-contextual spatial transformations destroy underlying data relationships, making image-based Transfer Learning architectures impractical for tabular tasks. Future work should focus on validating these results on large, raw datasets to ensure the high scores in the custom DNN were not influenced by multi-stage synthetic data artifacts, and exploring specialized tabular deep learning methods—such as feature embedding layers, attention-based networks, or tabular transformers—that respect the dimensional independence of features while leveraging deep learning strengths.

Acknowledgment

We thank Kaggle user arpan129 for publicly sharing the insurance fraud detection dataset used in this study. We would also like to thank our instructor, Dr. D. Sumathi, for her guidance throughout this project.

References

[1] Insurance Europe, ”The Impact of Insurance Fraud,” Technical Report, 2019. [Online]. Available: https://www.insuranceeurope.eu/mediaitem/0bf0af82-e7ef-4439-a763-d7862859d421/The%20impact%20of%20insurance%20fraud.pdf

[2] MStanford CS229 Student Project, ”Machine Learning in Fraud Detection,” 2018. [Online]. Available: https://cs229.stanford.edu/proj2018/report/261.pdf

[3] Paliwal et al. (2020), TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images. arXiv:2001.01469

[4] Chawla et al. (2011), TSMOTE: Synthetic Minority Over-sampling Technique. arXiv preprint: arXiv:1106.1813

[5] Dataset, https://www.kaggle.com/datasets/arpan129/insurance-fraud-detection