### **Comparison of VGG, TSSDNet, and STATNet for Audio Deepfake Detection**

| **Model/Technique** | **Detecting AI-Generated Speech** | **Real-Time Potential** | **Analysis of Real Conversations** |
| --- | --- | --- | --- |
| VGG (Feature-Based) | Uses handcrafted spectral features (MFCC, CQCC) for forgery detection. Works well but lacks adaptability. | Not real-time; relies on feature extraction & CNN classification. | Limited; trained on pre-recorded datasets, lacks robustness for live speech. |
| TSSDNet (End-to-End CNN) | Learns deep representations from raw audio using ResNet/Inception. Eliminates need for handcrafted features. | High real-time potential due to end-to-end processing on time-domain signals. | Strong; adaptable for live conversation analysis. |
| STATNet (Multi-Task Learning) | Performs spoof detection + source identification using spectral & temporal CNN features. | Moderate real-time potential, but computational cost is higher due to MTL. | Very strong; generalizes well across datasets & detects attack origins. |

### 

## **1. VGG (Feature-Based Forgery Detection)**

### Key Technical Innovation

### Uses precomputed spectral features (CQCC, MFCC) as input to a VGG-based CNN.

### Supervised learning model classifies speech as real or fake.

### Reported Performance

### ASVspoof2019: EER ~ 3-5% (CQCC + VGG).

### Works well on known synthetic speech techniques but lacks generalization.

### Why It’s Promising?

### Proven baseline with well-understood handcrafted features.

### Lower training cost than end-to-end models.

### Limitations

### Not real-time (requires preprocessing).

### Limited adaptability (fails on unseen attacks).

### Depends on handcrafted features, which may miss subtle deepfake artifacts.

### 

## **2. TSSDNet (End-to-End CNN for Live Detection)**

### Key Technical Innovation

### Uses ResNet/Inception-style CNN to learn features directly from raw waveforms.

### Eliminates handcrafted feature dependency.

### Reported Performance

### ASVspoof2019 → Better than state-of-the-art CQCC-based models.

### Strong cross-dataset generalization (e.g., ASVspoof2015).

### Why It’s Promising?

### Great for live detection; works directly on time-domain speech.

### Learns features automatically, making it more adaptive.

### Limitations

### Higher computational cost than feature-based methods.

### Potential overfitting if not trained on diverse data.

### 

## **3. STATNet (Multi-Task Learning for Robust Detection)**

### Key Technical Innovation

### Uses multi-task CNN architecture:

### Task 1 → Binary classification (Real vs. Fake).

### Task 2 → Source identification (Which TTS/VC model was used?).

### Learns both spectral and temporal features for better robustness.

### Reported Performance

### ASVspoof2019 LA → EER 2.456%.

### Cross-dataset (FOR-Norm, In-the-Wild Deepfake) → EER < 1% (high generalization).

### Why It’s Promising?

### Most robust approach; detects fake speech + identifies attack source.

### Strong cross-dataset generalization → useful in real-world conditions.

### Potentially detects new deepfake attack types before they become widespread.

### Limitations

### Higher computational cost than TSSDNet.

### Slower than real-time if not optimized.

### 

### **4. Final Recommendation**

### If real-time analysis is needed → TSSDNet.

### If robust, explainable detection is required → STATNet.

### If you need a fast but less adaptive approach → VGG-based models.

### 

### **Implementation Process Documentation**

**1. Challenges Encountered:**

* **Variable Spectrogram Dimensions:**
  + Audio files had different lengths, leading to mel-spectrograms with inconsistent time frames.  
    **Resolution:** Implemented padding/truncation to fix the spectrogram shape (e.g., 128×128 or 224×224) so that all inputs are uniform.
* **Channel Mismatch for Pretrained Models:**
  + Pretrained VGG16 expects 3-channel (RGB) images, but mel-spectrograms were initially generated as single-channel.
  + **Resolution:** Converted the single-channel spectrograms into 3-channel images by repeating the channel.

**2. How Challenges Were Addressed:**

* Adjusted spectrogram extraction parameters and implemented a pad\_or\_truncate function to ensure a consistent input size.
* Modified data preprocessing to convert grayscale spectrograms into 3-channel images, aligning with VGG16’s input requirements.

**3. Assumptions Made:**

* **Fixed Input Dimensions:**
  + Assumed that a fixed spectrogram size (e.g., 224×224) is sufficient to capture relevant audio features for classification.
* **Upsampling/Interpolation:**
  + Assumed that directly generating mel-spectrograms at the desired resolution is preferable over resizing lower resolution outputs to avoid interpolation artifacts.
* **Pretrained Model Adaptation:**
  + Assumed that fine-tuning a VGG16 model pretrained on ImageNet is effective for audio deepfake detection once the input format (3-channel images) is matched.
* **Subset Evaluation:**
  + Assumed that evaluating the model on a random subset of 1,000 rows from a larger dataset is representative of its generalization capability on the In-the-Wild dataset.

**4. Datasets Used:**

* **Training:** CMFD dataset containing both English and Chinese audio files (with tampered and untampered classes).
* **Testing (Generalization):** In\_the\_wild dataset with artificially generated audio files from RawNet 2, RawGAT-ST, and PC-Darts, along with their real counterparts. (only english)

5. **Analysis: Why This Model Was Selected**

* **Proven Architecture:** VGG16 is a well-established convolutional neural network known for its robust feature extraction capabilities. Its architecture has been widely used and validated in image classification tasks, and by converting mel-spectrograms to 3-channel images, we can leverage these strengths for audio deepfake detection.
* **Transfer Learning Benefits:** Using a model pretrained on ImageNet allowed us to take advantage of transfer learning, significantly reducing the training time and computational resources needed compared to training a model from scratch.
* **Resource Constraints:** Due to limited computational resources, a lightweight and efficient solution was necessary. VGG16’s relatively straightforward architecture made it a practical choice that balanced performance and resource efficiency.

**6. Model working**

We implemented the **VGG16** model for audio deepfake detection by treating **Mel-spectrograms** as images. VGG16 is a deep convolutional neural network known for its simplicity and effectiveness in image classification tasks.

Our implementation steps:

1. **Input Layer** – The input Mel-spectrograms were resized to **224×224×3** to match the expected input shape of VGG16 (originally trained on ImageNet RGB images).  
    *(Resizing allows compatibility; 3-channel duplication was used since VGG expects 3 channels)*
2. **Convolutional Layers** – VGG16 consists of 13 convolutional layers.  
    *(Each* ***Conv layer*** *applies a set of filters via* ***2D convolution (dot product + bias)*** *followed by a* ***ReLU activation (f(x) = max(0, x))****, extracting increasingly complex features from the input)*
3. **Max Pooling Layers** – After every 2 or 3 convolutional layers, a **MaxPooling layer** is applied.  
    *(This layer reduces the spatial size using a* ***max operation*** *over a 2×2 window, which helps downsample features and makes the model more efficient and robust to translations)*
4. **Flatten Layer** – Converts the 3D feature maps into a 1D feature vector.  
    *(This is a reshaping operation with no learning, necessary before passing data to dense layers)*
5. **Dense Layers** – Fully connected layers process the flattened features.  
    *(Each* ***Dense layer*** *performs a* ***matrix multiplication plus bias****, followed by a* ***ReLU activation****, capturing non-linear combinations of features)*
6. **Dropout Layer** – Added between dense layers to prevent overfitting.  
    *(It randomly sets a fraction of input units to zero during training; mathematically, a* ***Bernoulli mask*** *is applied to each unit)*
7. **Final Output Layer (Dense)** – A single neuron with **sigmoid activation** was used for binary classification.  
    *(Outputs a probability using* ***sigmoid: f(x) = 1 / (1 + exp(-x))****, representing the likelihood of tampered or untampered audio)*

### ***Performance Results***

* ***Training/Test on CMFD Dataset***
  + ***Accuracy****: 62.75%*
  + ***AUC-ROC****: 0.693*
  + ***Confusion Matrix****:  
     [[219 181]  
     [117 283]]*
* ***Generalization Test on In-the-Wild Dataset***
  + ***Accuracy****: 52.3%*
  + ***AUC-ROC****: 0.618*
  + ***Confusion Matrix****:  
     [[254 365]  
     [112 269]]*

### ***Observed Strengths***

* *The model captures important discriminative features from Mel spectrograms.*
* *Demonstrates above-baseline generalization, even to unseen, real-world audio synthesis methods.*

### ***Observed Weaknesses***

* *Notable drop in performance on the In-the-Wild dataset suggests domain sensitivity.*
* *Likely affected by broader variation in speakers, background conditions, and synthesis techniques used in the wild dataset.*
* *High false positive rate indicates difficulty in distinguishing real and spoofed voices when the spoofing methods are unfamiliar.*

### ***Suggestions for Future Improvements***

* ***Incorporate LSTM layers*** *to capture temporal dependencies in audio. This can be added on top of the existing CNN feature extractor without starting from scratch, enabling the model to learn sequential patterns over time frames.*
* ***Use data augmentation*** *to simulate real-world distortions and voice variations.*
* ***Explore domain adaptation methods*** *to align feature distributions across datasets.*