Material Identification with mmWave Vibrometry and Vision.

Charchit Gupta*, *International Institute of Information Technology, Hyderabad, India

I. BACKGORUND

The foundation of this project rests upon the innovative RFVibe system, designed for contactless material and object identification through the fusion of millimeter wave wireless signals and acoustic signals, obviating the need for RFID tags. RFVibe operates by inducing micro-vibrations in objects through audio sound, subsequently captured and analyzed by a millimeter wave radar to extract their vibrational frequencies. The pursuit of enhancing RFVibe's accuracy and capabilities, particularly through the integration of vision, underscores the motivation for this project. The incorporation of vision-based improvements aims to elevate RFVibe's performance and broaden its applicability in scenarios demanding precise and comprehensive object identification.

II. DESIGN

A. Mask

1) Prior Works: In the pursuit of effective object segmentation, three notable methods have been considered. GrabCut, a traditional approach, has provided decent results but is considered somewhat outdated (Rother et al., 2004). A more recent advancement, Detectron2, demonstrates improved performance when initialized with bounding boxes; however, it yields multiple object masks, which may be undesirable. Moreover, its limitations become evident in handling transparent objects and those not encountered during training (Wu et al., 2019). In contrast, FAIR's Segment Anything model, a cutting-edge solution, outshines its counterparts. With superior performance when initialized with bounding boxes and input points, it not only surpasses GrabCut and Detectron2 but also **excels in handling transparent objects**. Consequently, the choice for object segmentation in this study aligns with FAIR's Segment Anything model for its robust performance and versatility (FAIR, 2023).

2) SAM Refinement: In refining FAIR's Segment Anything model for our experiments, a crucial modification involved consistently providing the model with three specific inputs (for each dataset): bounding boxes delineating target objects, foreground points indicating regions of interest, and background points characterizing non-object areas. The bounding box should be big enough to include the bigger objects and the foreground input points should be selected according to the smaller object positions. This deliberate modification aims to ensure a consistent exposure of the model to essential contextual cues, facilitating a robust and comparable assessment of its adaptability and efficacy in various scenarios.

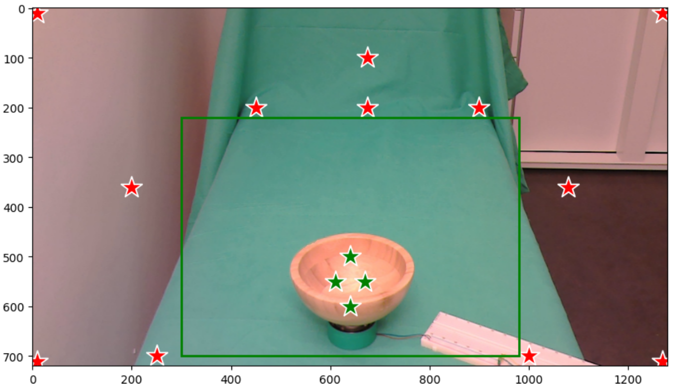


Fig. 1: SAM Refinement

B. Neural Network

1) Mask Feature: In the generation of the intermediate mask feature map, a convolutional neural network (CNN) is employed to process the downsampled mask, ensuring efficient memory consumption. The mask, reduced by a factor of 8 to dimensions of 160×90, undergoes convolutional operations inspired by the architectures of Faster R-CNN (Ren et al., 2015) and Mask R-CNN (He et al., 2017), as elaborated in the figure 2. The CNN employs a kernel size of 3, a stride of 2, and padding of size 1. At each stage, convolution, batch normalization, and ReLU activation operations are applied, collectively forming a 128-dimensional vector intermediate feature map.

2) Feature Heads: RFVibe's neural network incorporates four distinct feature heads, each tailored to process specific input features and contribute to a shared latent space:

- **Frequency Feature Head:** Takes a 1×286 input feature and outputs an 8×128 intermediate feature map through 8 convolutional layers.
- **Power Feature Head:** Processes a vector of length 25, utilizing two stacked fully connected layers with max-pooling and dropout to output a 1×128 feature map.
- **Damping Feature Head:** Mirrors the architecture of the Frequency Feature Head but takes inputs of size 8×125 and outputs 8×128-sized intermediate feature maps.
- **Mask Feature Head:** Takes in a mask, a 2D vector of shape 160×90, and employs a convolutional neural network to output a 1×128 intermediate feature map.

These feature heads collectively create a common space where intermediate features are either summed or concatenated. Subsequently, the classification heads process these

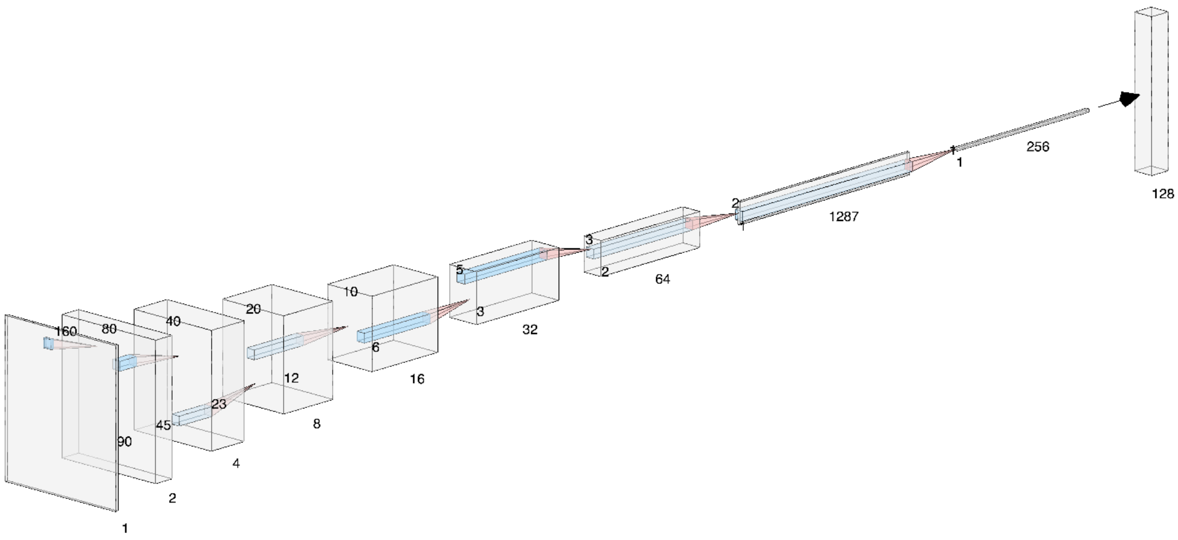


Fig. 2: Convolutional Neural Network design for the Mask Feature

intermediate feature maps to classify into one of N possible classes, maintaining a consistent architecture. The final stage involves the aggregation head, which combines intermediate feature maps from all sources.

III. Implementation

A. Hardware Setup

RFVibe uses TI's IWR1843BOOST evaluation module with the DCA1000EVM. Two transmitters and four receivers with a two-dimensional virtual array were used to capture data at 77 GHz. The FMCW chirp was designed to have a frame rate of 250Hz. The chirp parameters created a range resolution of 5.63 centimeters, a maximum range of 3.24 meters, and had a sweep bandwidth of 2.6 GHz. Received data was sent to a host PC using mmWave Studio. A SparkFun Surface Transducer kept directly under the target object, was used to produce micro-vibrations.

B. Software Setup

We modified the OpenRadar GitHub to connect the mmWave Studio and scripting for our experiment setup. We calibrated the raw data in MATLAB, and used Python to extract the features used.

C. Transducer Setting

In the experimental setup, the surface transducer is powered by a pulse generator configured with a 50% duty cycle and maximum input voltage. This specific setting orchestrates a controlled 'tapping' motion onto the object, producing micro-vibrations within the target object.

D. Neural Network Parameters

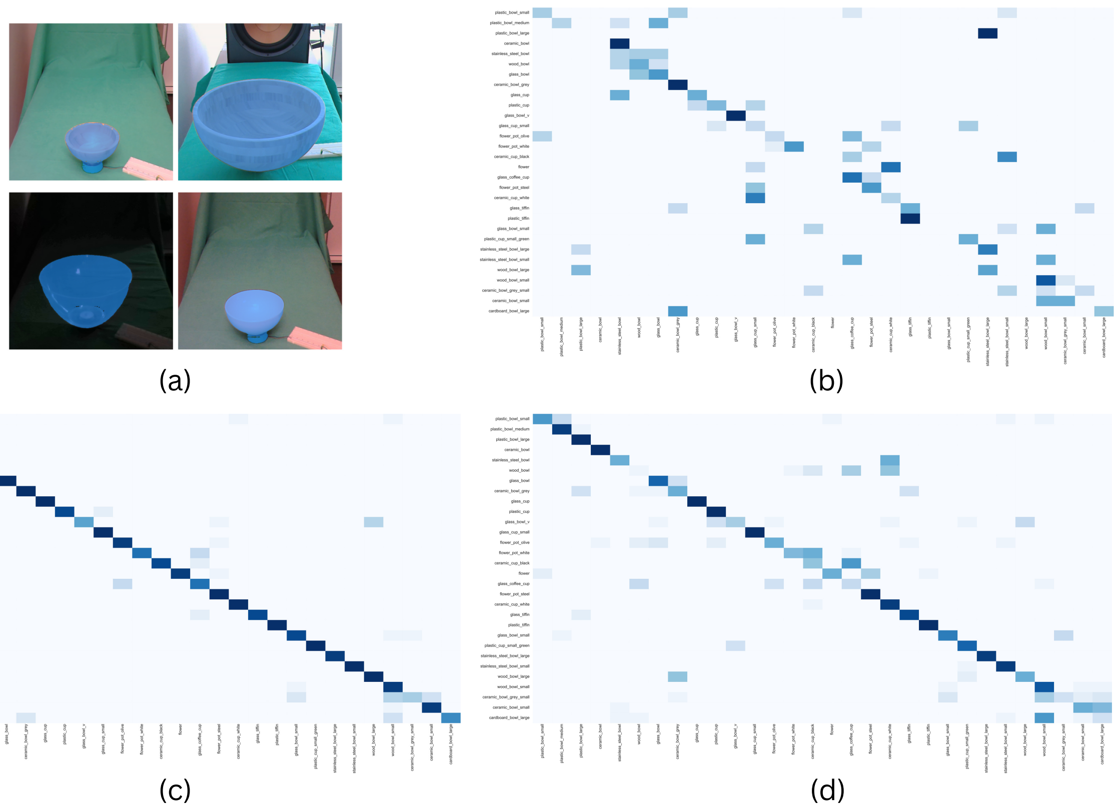
RFVibe is designed with PyTorch. It uses a Adam optimizer with learning rate 0.001. The loss is calculated using cross-entropy loss. The batch size is 8, and the dropout probability is 0.25. After being fed to RFVibe all three features are resized to size 128 for each of the intermediate layers. The final loss weights are set to $(W_1, W_2, W_3, W_4, W_5)$ = (0.9, 0.3, 0.3, 0.1, 1).

E. Experiment Environment

The data collection setup took place in one room only where the laptop, radar, and the object were kept on different cabinets, various environment changes were made to ensure a diversified dataset. The radar position and the spacing between the radar and object cabinets was changed for each dataset. Each experiment ran for 13 seconds, during 10 seconds of which the speaker was playing the sound source. The object was positioned on top of the transducer, 20 - 50 cm away from the radar. Experiments were performed in an imprecise manner to mimic everyday use. This meant objects were not placed in an exact location or orientation to the radar. The surroundings of the experiments had various amounts of background noise and movement. In addition, the point of contact of the target object with the transducer also varied greatly, which affected the visibility of certain frequencies, which made sure that the network did not overfit to a specific volume.

F. Datasets

The dataset includes 30 different objects of various shapes and sizes consisting of the six materials. The objects include plastic bowls (s/m/l), white ceramic bowls (s/m), stainless steel bowls (s/m/l), wood bowls (s/m/l), glass bowls (m/l), grey ceramic bowls (s/m), glass cup (s/m), plastic cup (s/m), flower pots (olive/white/black/steel), ceramic cups (black/white), glass coffee cup, tiffins (glass/plastic), green plastic cup (small), cardboard bowl (large). Similarly, the material distribution includes ceramic (8 objects, 1698 instances), metals (4 objects, 855 instances), plastic (7 objects, 1489 instances), glass (7 objects, 1491 instances), wood (3 objects, 650 instances), and cardboard (1 objects, 230 instances). Note



(a) (b) (c) (d)

Fig. 3: (a) Masks for objects: WBS, WBL, PBS, CBS; Confusion matrices: (b) Only Mask, (c) RFVibe with mask, (d) RFVibe w\o mask

that *these objects are particularly smaller than the ones used for RFVibe testing*. The dataset was split into training, validation and test sets. The validation set was approximately 43% of the training set size, and the test set was approximately 14% of the training set size. Experiments were split into three datasets such that they had no overlap in experiment runs. Between train, validation and test sets, there are no experiments done in the exact same environment. An experiment run includes a set of 10 experiments per object where the physical setup did not change. While the power feature for these experiments remains relatively similar between the 10 experiments, the objects vibration features still fluctuates. After each run, the radar was turned off, and the physical set-up was perturbed. For the results in Sec. 4 the dataset we used has 3985, 1708, and 599 experiments for the training, validation, and test split, respectively.

IV. Results

A. Mask

Post-refinement, FAIR's Segment Anything model demonstrated robust performance, given that the green screen did not entirely cover the background. With roughly optimal input refinement parameters, the model had an object detection accuracy between 96% and 100%. Notably, there were instances where the generated mask inadvertently included sections of the breadboard, particularly the black holes, owing to similarities in texture and color with the target object.

B. Object Classification

The presented model achieves an overall accuracy of 83.0% in object classification, as illustrated by the detailed confusion matrix for all 30 objects in Fig. 3(c). When evaluating the model on the same dataset without the inclusion of the mask, the overall accuracy decreases to 68%, as depicted in Fig. 3(d). This signifies an improvement of 15% in accuracy when incorporating the mask feature. The dedicated confusion matrix for the mask feature alone is exhibited in Fig. 3(b). Clearly, the mask is not powerful enough to classify each object on its own, a property which is very essential. The mask encounters challenges in distinguishing objects with similar shapes, exemplified by the confusion between stainless steel bowl, wood bowl, and glass bowl. Other instances of misclassification include the plastic tiffin being mistaken for the glass tiffin and the confusion between large plastic and stainless steel bowls. Notably, the large glass bowl, distinguished by its distinct 'V' shape, achieves 100% classification accuracy based solely on the mask. This underscores the need of the mask feature in scenarios where objects share similar fft, damp, and mrf_squared features but differ significantly in shape, as evidenced by the substantial enhancement in the confusion matrix upon incorporating the mask.

C. Material Classification

The effectiveness of our material classification hinges on a well-distributed representation of objects across various material classes within our dataset. Specifically, our dataset comprises 1 cardboard object, 3 wood objects, 4 steel objects, and at least 7 objects of other material types. Notably, the lone cardboard object exhibited the third least accurate object classification accuracy, and the three wood objects showcased diverse frequency features, further deteriorating the material classification. When considering all objects, our model achieves a material classification accuracy (MCA) of 54%. An interesting observation is that the MCA for materials aligns proportionally with their respective sample sizes — cardboard and wood exhibit MCAs of 5% and 20%, respectively, while other materials have accuracies exceeding 78%. Upon excluding cardboard and wood, our model achieves an accuracy of 74.6%. Further eliminating glass results in an accuracy boost to an impressive 92%, with ceramic having a MCA of 100%. Clearly, a more balanced dataset is imperative for attaining enhanced material classification accuracy.

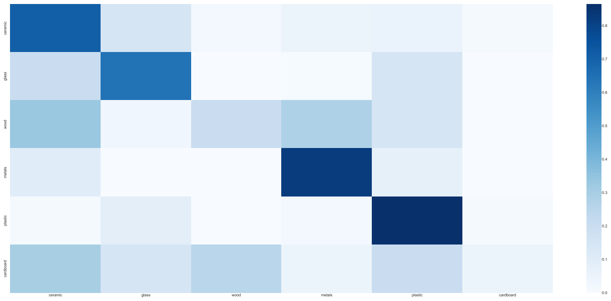


Fig. 4: Confusion Matrix for materials

D. Microbenchmarks

1) Frequency Feature vs Average STFT: The frequency feature encompasses 5 windowed FFTs of overlapping phase windows, while the average Short-Time Fourier Transform (STFT) feature represents the squared average of the STFT feature over time. Following our intuition, which suggests that the latter should contain less information than the former, our results show a noticeable decline in overall object accuracy. Specifically, the accuracy drops to 69%, a decrease from the 83% achieved by the original Frequency feature.

2) Transducer on the cabinet: When the transducer on the cabinet was positioned upside down, it failed to generate the expected micro-vibrations in the object. The location of the transducer proved crucial; when placed at a distance, no vibrations were observed. However, placing it closer revealed some micro-vibrations, yet the vibrations from the cabinet seemed to dominate. Interestingly, the large steel bowl exhibited discernible frequency peaks. Additionally, aligning the transducer perpendicular to the radar direction yielded slightly improved results. Unfortunately, the non-reproducible nature of these frequency plots led us to dismiss further consideration of this method.

3) Transducer taps from the side: An improved yet impractical method involved placing the transducer on the table, ensuring its face made contact with the object from the side. While this approach mirrored the 'tap' experiments, its execution posed challenges. Specifically, the distance between the object and the transducer continually increased over time due to collisions between them, making the process cumbersome.

4) Transducer Platform: A more practical approach involved placing a piece of cardboard on top of the transducer, serving as a platform for the objects. While this method was convenient, it proved ineffective in generating micro-vibrations in the objects. Surprisingly, the micro-vibrations from the cardboard appeared to dominate the frequency feature, evident from consistent strong frequency peaks around 30Hz for all tested objects. These peaks overwhelmed any other frequency characteristics, indicating a limitation in the method's ability to isolate object-specific vibrations.

5) Mask Flipping: To enhance the dataset's diversity, we conducted Mask Flipping, involving the flipping of masks along the vertical axis. This introduces randomness in object positions, preventing the neural network from overfitting to specific object positions during data collection. Despite the absence of apparent overfitting in the results, there remains potential for exploring additional modifications to the mask for further experimentation.

6) Pulse Generator Configurations: To ensure effective generation of micro-vibrations, the pulse generator's voltage was kept at the maximum level due to insufficient performance at lower voltages.

Regarding the duty cycle, the time required for micro-vibrations to damp varies based on the object's characteristics. Larger stainless steel bowls, for instance, demand more time compared to smaller plastic bowls. A duty cycle of 50 proves to be an optimal choice as increasing the duty cycle limits the time available for micro-vibrations to damp, while decreasing it results in insufficient tap power duration due to a reduction in overall tap power. The 50% duty cycle strikes a balance in achieving optimal micro-vibration characteristics.

7) Object to radar perpendicular distance: The 2D antenna array of our mmWave device was leveraged to beamform and zoom into the angles with the strongest reflection. Despite the effectiveness of beamforming, challenges emerged when dealing with smaller objects like bowls when their distance from the radar line was increased (For most cases, it was kept near 0).

8) Obect to Radar distance: The closer, the better. The frequency feature took a hit for smaller bowls when the objects were placed far from the radar (>60 cm). Thus, even with beamforming, the curved shape caused problems. For that reason, the dataset restricts the object distance to 50 cm.