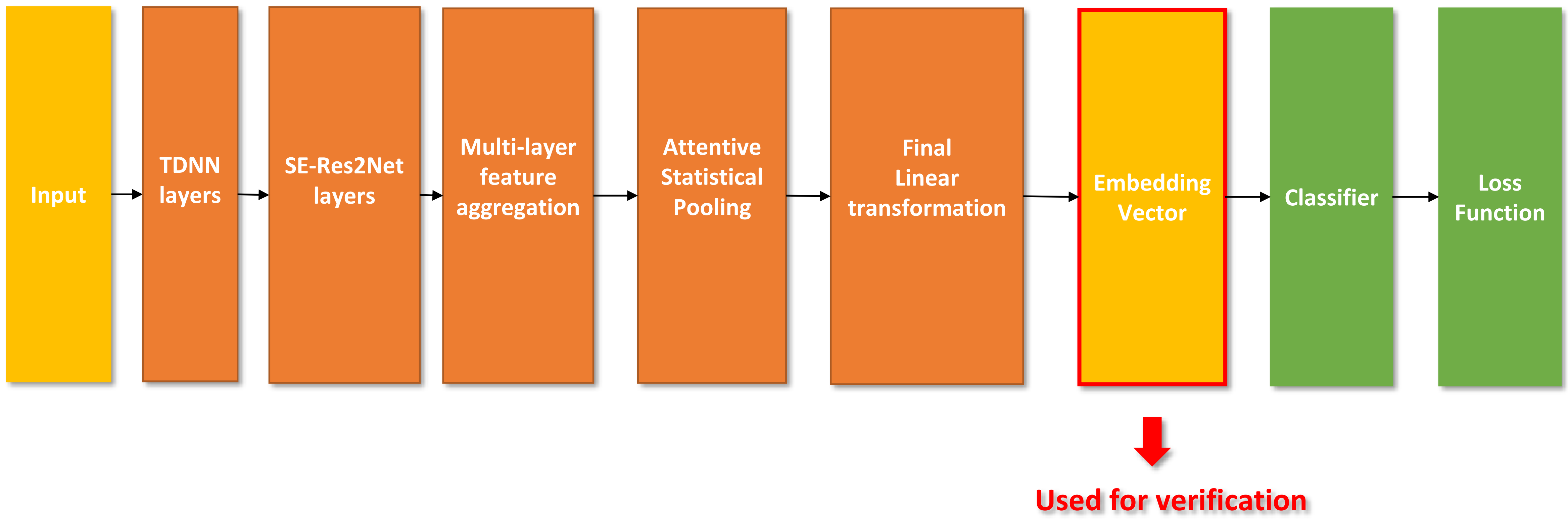
Speaker Verification

Using spkrec-ecapa-voxceleb

Model

- Speaker Verification with ECAPA-TDNN embeddings on Voxceleb
- Text Independent Speaker Verification (TI-SV) Model
- performance on Voxceleb1-test set(Cleaned): 0.8% EER
- Structure:
    - composed of an ECAPA-TDNN model
    - embeddings are extracted using attentive statistical pooling
    - trained with Additive Margin SoftMax Loss
    - Verification using cosine distance between speaker embeddings


Input
TDNN layers
SE-Res2Net layers
Multi-layer feature aggregation
Attentive Statistical Pooling
Final Linear transformation
Embedding Vector
Classifier
Loss Function
Used for verification

Loss Function

- Additive Margin SoftMax Loss
- Goal:
  - Set margin between classes

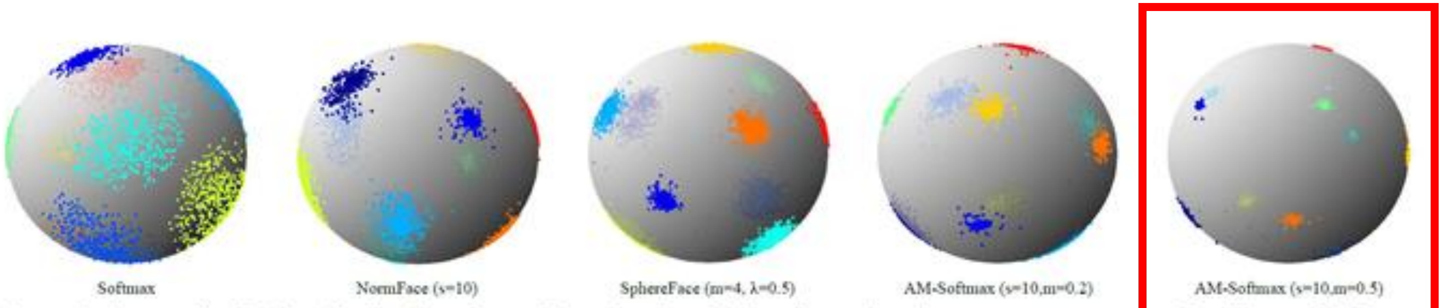


Figure 4. Feature distribution visualization of several loss functions. Each point on the sphere represent one normalized feature. Different colors denote different classes. For SphereFace [9], we have already tried to use the best hyper-parameters we could find.

Feature Vector Normalization

- Normalize feature embedding vector before calculating loss
- In face recognition
    - data with lower vector norm works better without feature normalization
- In speaker verification
    - Clean audio has lower vector norm than noise added audio
- Test on clean audio:
    - Verification using model with & without feature normalization
    - result:
        - w/o feature normalization doesn’t work better

Final Goal

- Run on smartwatch GPU (need smaller model)
- Good performance with various enrollment phrases different from training data

Experiments

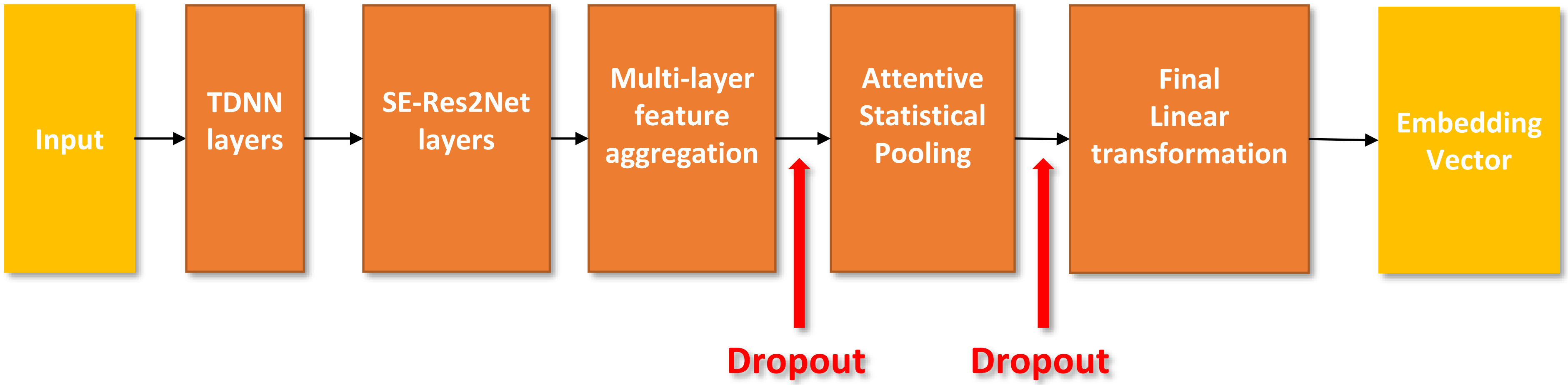
- Train Data:
  - Hi-MIA dataset (9 speakers, 7204 utterances)
  - Speech Command dataset (1590 speakers, 3941 utterances)
- Test Data: 1000 trials per test
- Training Time: 8~12 min per epoch
- Criteria: EER (equal error rate)

1. Training & Testing Languages

Train Data/ Test Data	“Yes” (English)	“Hi Mia” (Chinese)
“Yes”	0.08	0.11
“Yes” + “Hi Mia”	0.08	0.08

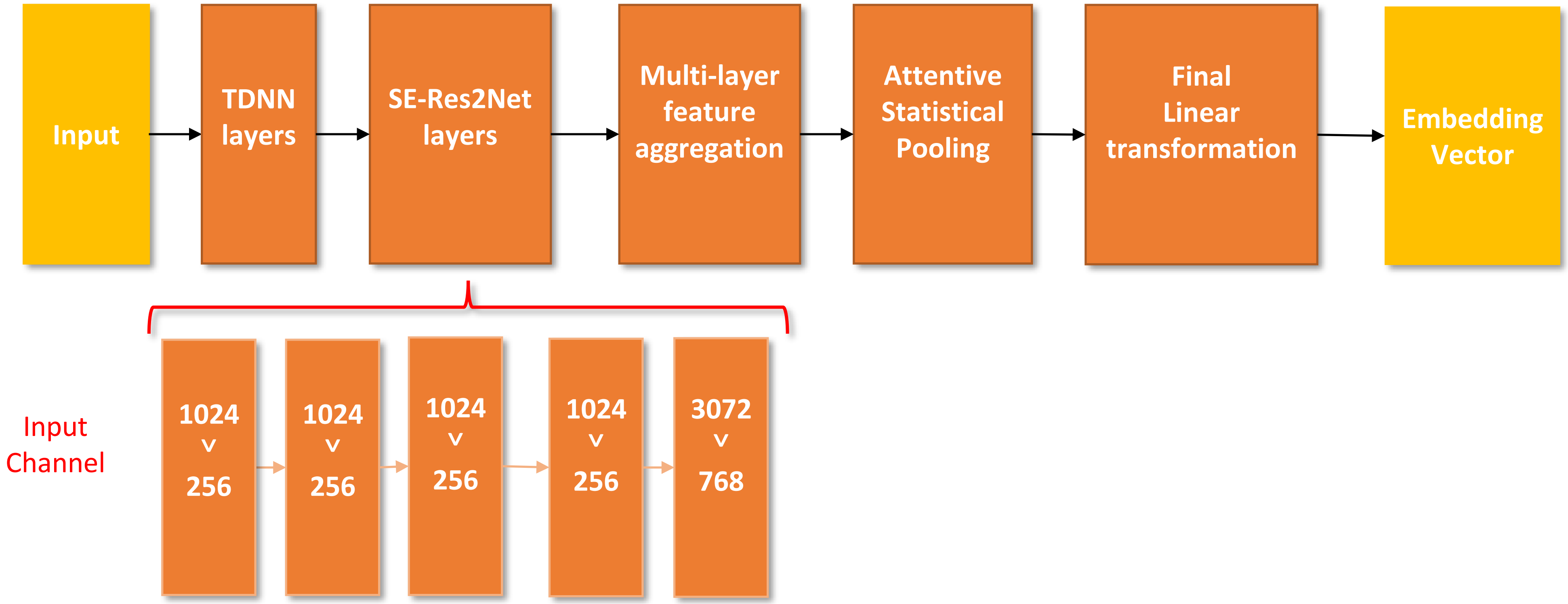
2. Dropout Layers

	"Yes"	"Hi Mia"	"off"	"marvin"
original	0.08	0.08	0.05	0.05
2 dropout layers	0.12	0.11	0.06	0.08



4. Model Size

	“Yes”	“Hi Mia”
Original	0.08	0.11
Smaller SE-Res2Net layers	0.09	0.22



5. Different Enrollment and Test Text

	Enroll: “off” Test: “left”	Enroll: “down” Test: “stop”	Enroll: “up” Test: “down”	Enroll: “off” Test: “off”	Enroll: “Marvin” Test: “Marvin”
original	0.11	0.17	0.12	0.05	0.06
With dropout layer	0.11	0.14	0.07	0.06	0.08
Smaller Model	0.10	0.25	0.11	0.06	0.07

5. Speed

Model	Average Verification Speed (sec)
Original	0.11
With dropout	0.09
Smaller model	0.03

Conclusion

- Training with language same as testing data improves performance
- Adding dropout layer doesn’t decrease EER
- Need more training data
- Smaller model
  - converges
  - bad performance on some text