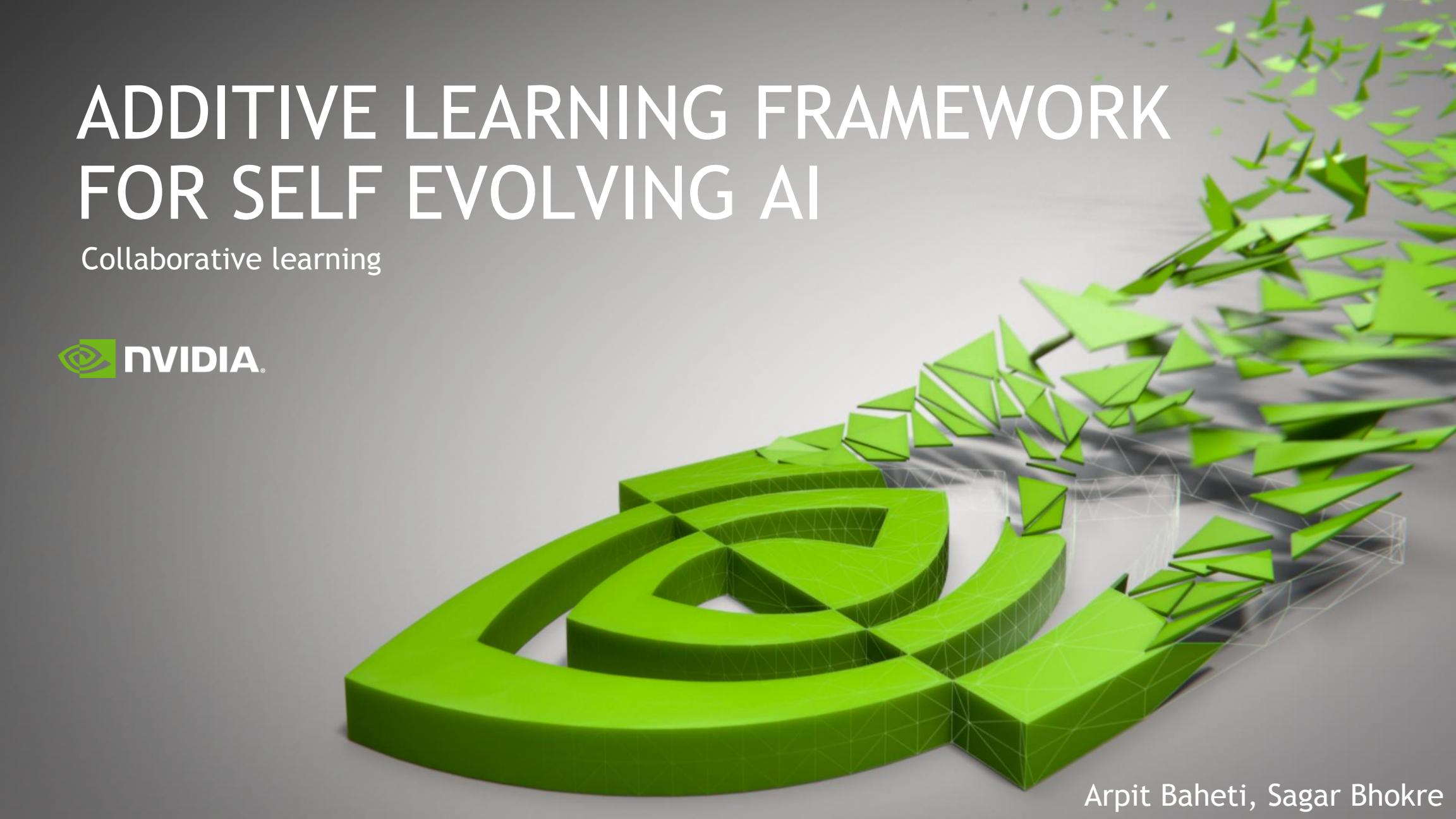
ADDITIVE LEARNING FRAMEWORK FOR SELF EVOLVING AI

Collaborative learning



Arpit Baheti, Sagar Bhokre

OBJECTIVES

Demonstrate additive learning for modules estimating User Identity

---

- What is additive learning?
  - AI modules helping each other for improved confidence
- Why additive learning?
  - Because networks need to evolve and stay up-to-date
- Where can these be employed?
  - In homes, cars, where user interacts frequently

OBJECTIVES

Demonstrate additive learning for modules estimating User Identity

---

- How are these useful?
  - correct functionality with changing subject features
- When to use it?
  - Low risk local system updates fine tuned for user experience

HOW IT WORKS

Strategies

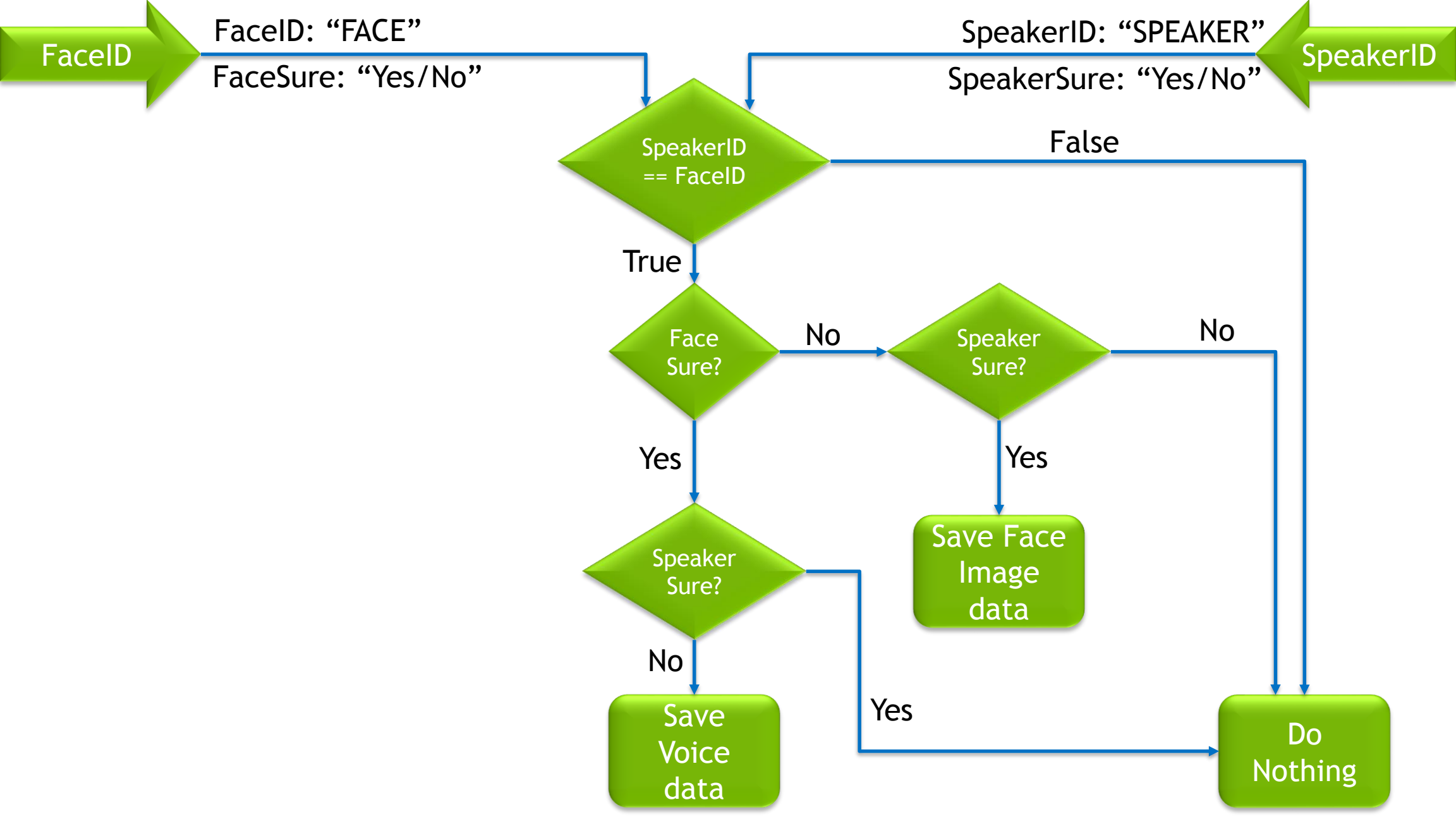
- *Enrollment phase:*
  - Enroll subject into the system
- *Selection phase:*
  - Select entries from stored database
- *Evolution phase:*
  - Retrain the classifier with selected entries from above phase

ENROLLMENT PHASE

ENROLLMENT STRATEGY

- Collect data (Face and Voice) for enrolling the subject, using camera and mic
- Generate embeddings for enrolled person
- Train classifier using generated embeddings

SELECTION PHASE


FaceID
FaceID: “FACE”
FaceSure: “Yes/No”
SpeakerID: “SPEAKER”
SpeakerSure: “Yes/No”
SpeakerID
SpeakerID == FaceID
False
True
Face Sure?
No
Speaker Sure?
No
Yes
Yes
Speaker Sure?
Save Face Image data
No
Save Voice data
Yes
Do Nothing

SELECTION STRATEGY

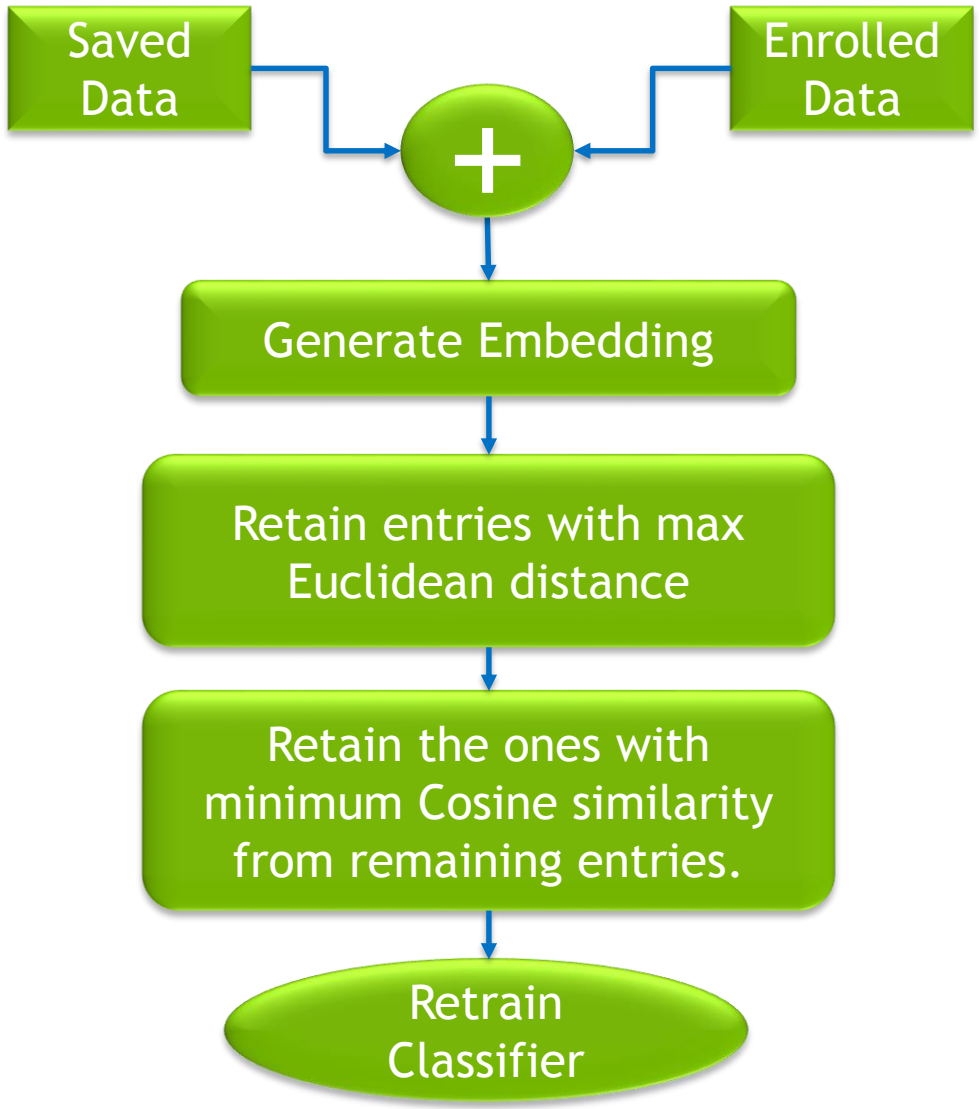
	FACE-ID SURE	SPEAKER-ID SURE	SAVE
Case I	No	No	None
Case II	Yes	No	Voice data
Case III	No	Yes	Image data
Case IV	Yes	Yes	None

EVOLUTION PHASE

EVOLUTION STRATEGY

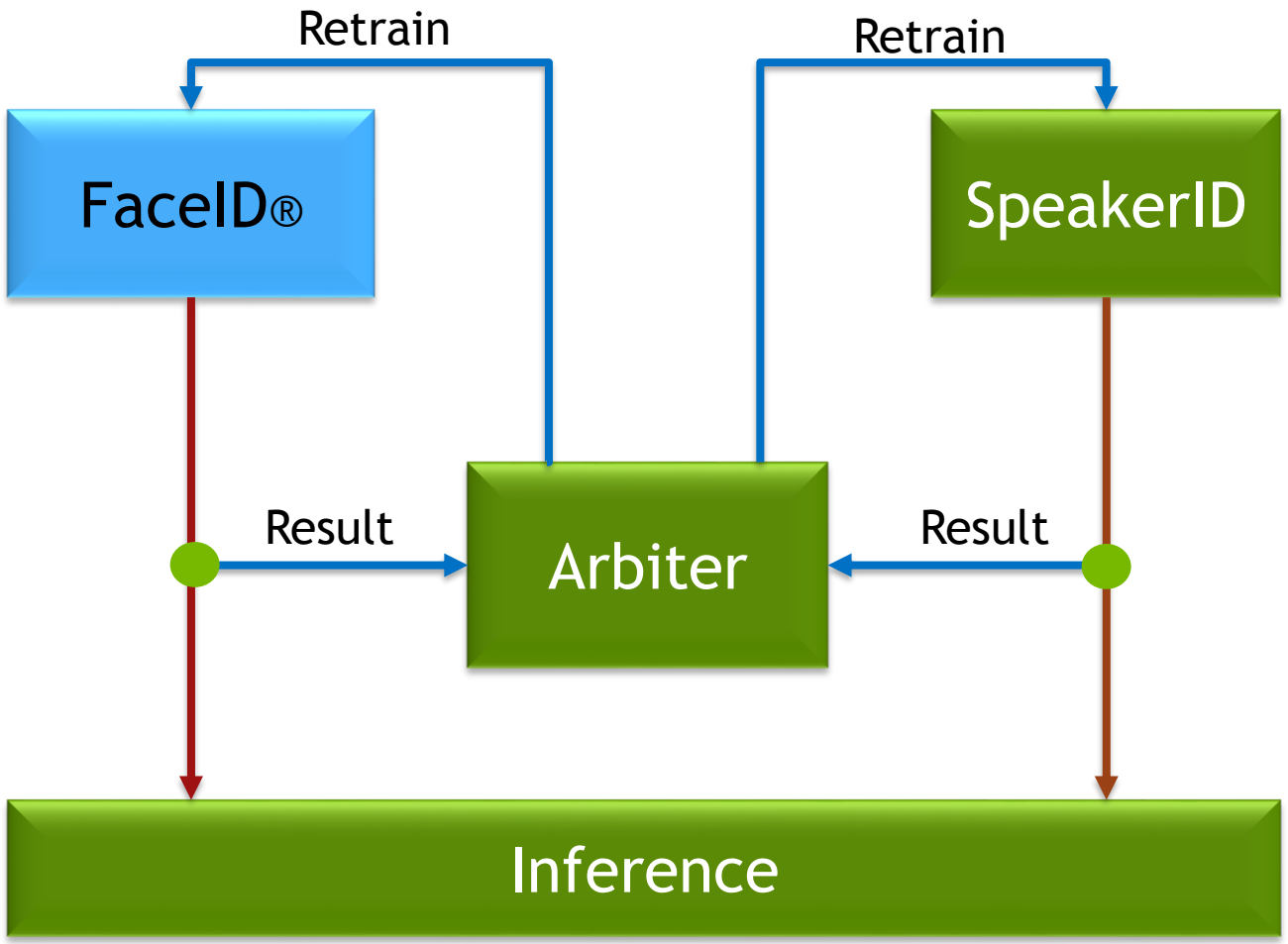
- Retrieve all (enrolled + saved) dataset.
- Generate embeddings for each data point.
- Select diverse Images/Audio to be used to retained using Euclidean and Cosine filters.
- Retrain if new information is added.

EVOLUTION STRATEGY



IMPLEMENTATION DETAILS

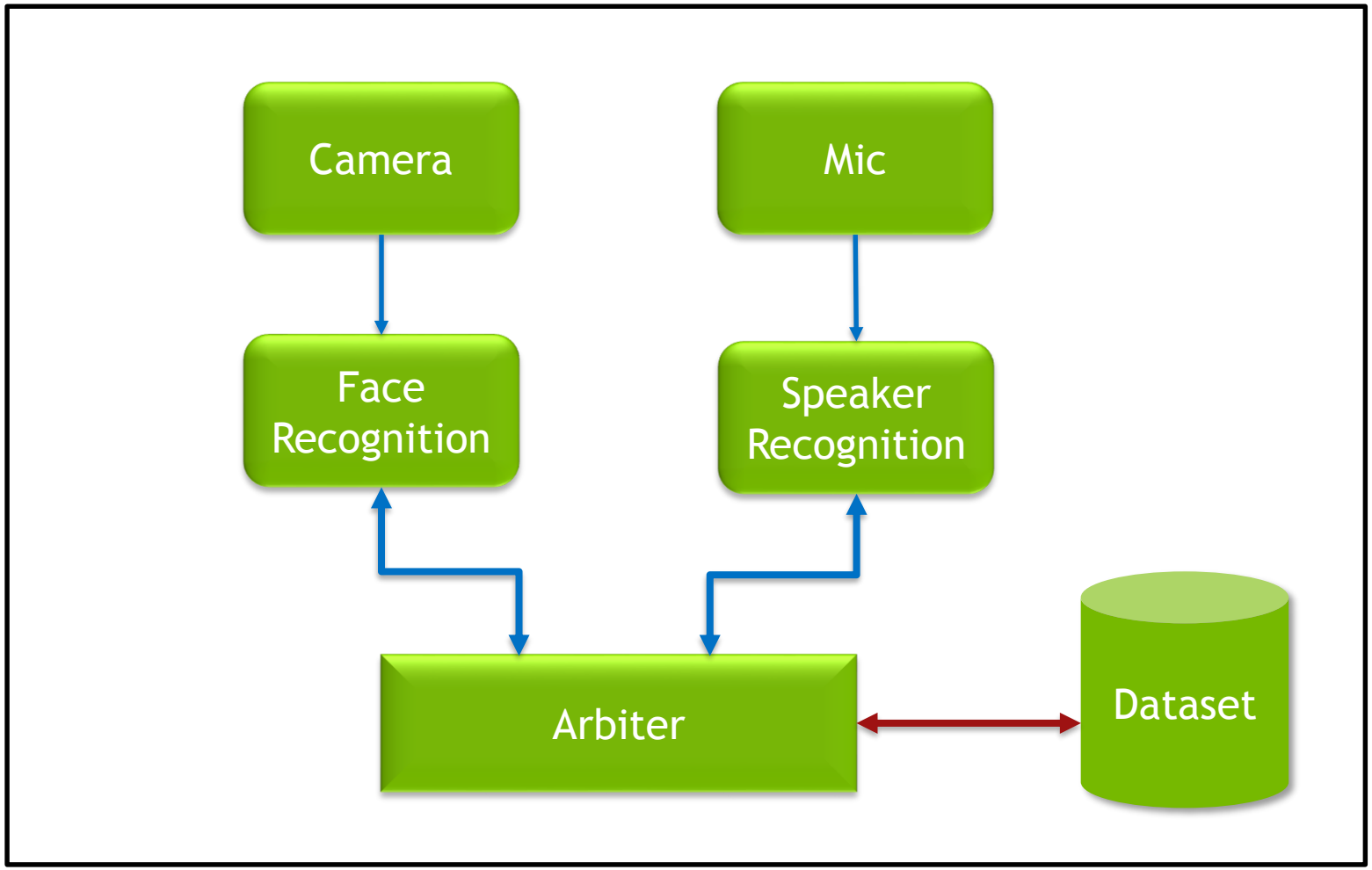
ARCHITECTURE



® https://cmusatyalab.github.io/openface/

Developed

WORKING PIPELINE



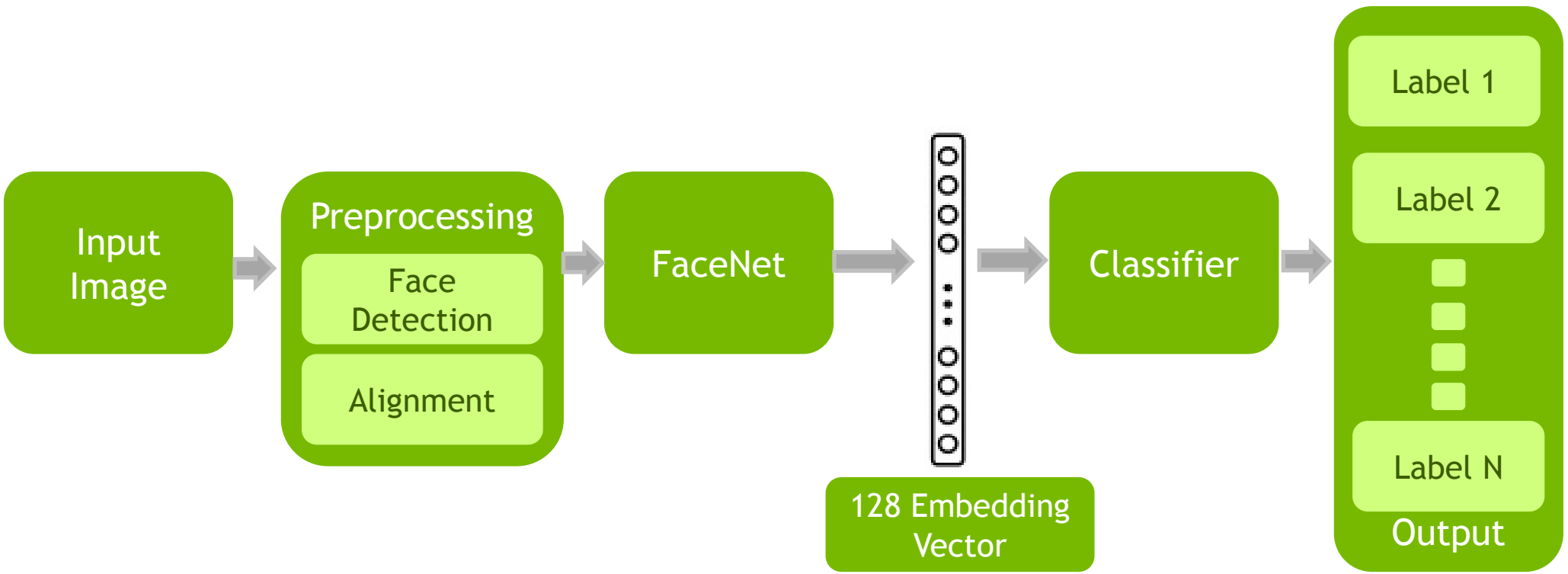
ENROLLMENT

- OpenFace for generating Face embeddings
- SpeakerNet for generating Voice embeddings
- RadialSVM classifier for classification Training

FACE RECOGNITION

- OpenFace generates embedding to train classifier.
- FaceID module uses OpenFace’s confidence to generate surety.
- Surety is calculated based on classifier confidence, Euclidean distance and cosine similarity.

FACE RECOGNITION

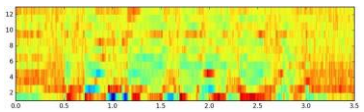


Reference: https://cmusatyalab.github.io/openface/

SPEAKER RECOGNITION

- SpeakerNet Deep Neural Network generates embedding.
- RadialSVM classifier uses embedding to generate labels and confidence.
- Surety is calculated based on classifier confidence, Euclidean distance and cosine similarity.

SPEAKERNET



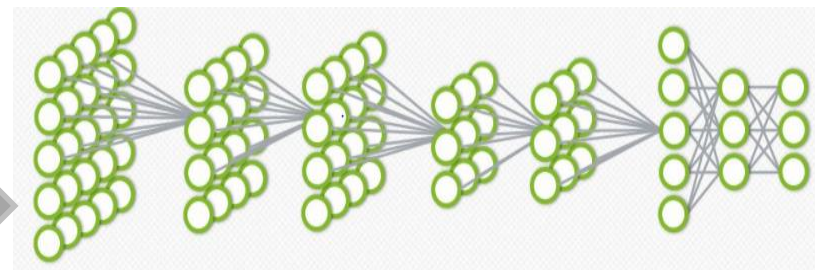
Input Audio

Preprocessing

- Noise Removal
- Silence Removal

Feature Extraction

64 MFCC Feature
1st Δ
2nd Δ



Convolutional Deep Neural Network

Triplet Loss

512 embedding Vector

SPEAKERNET TRAINING DETAILS

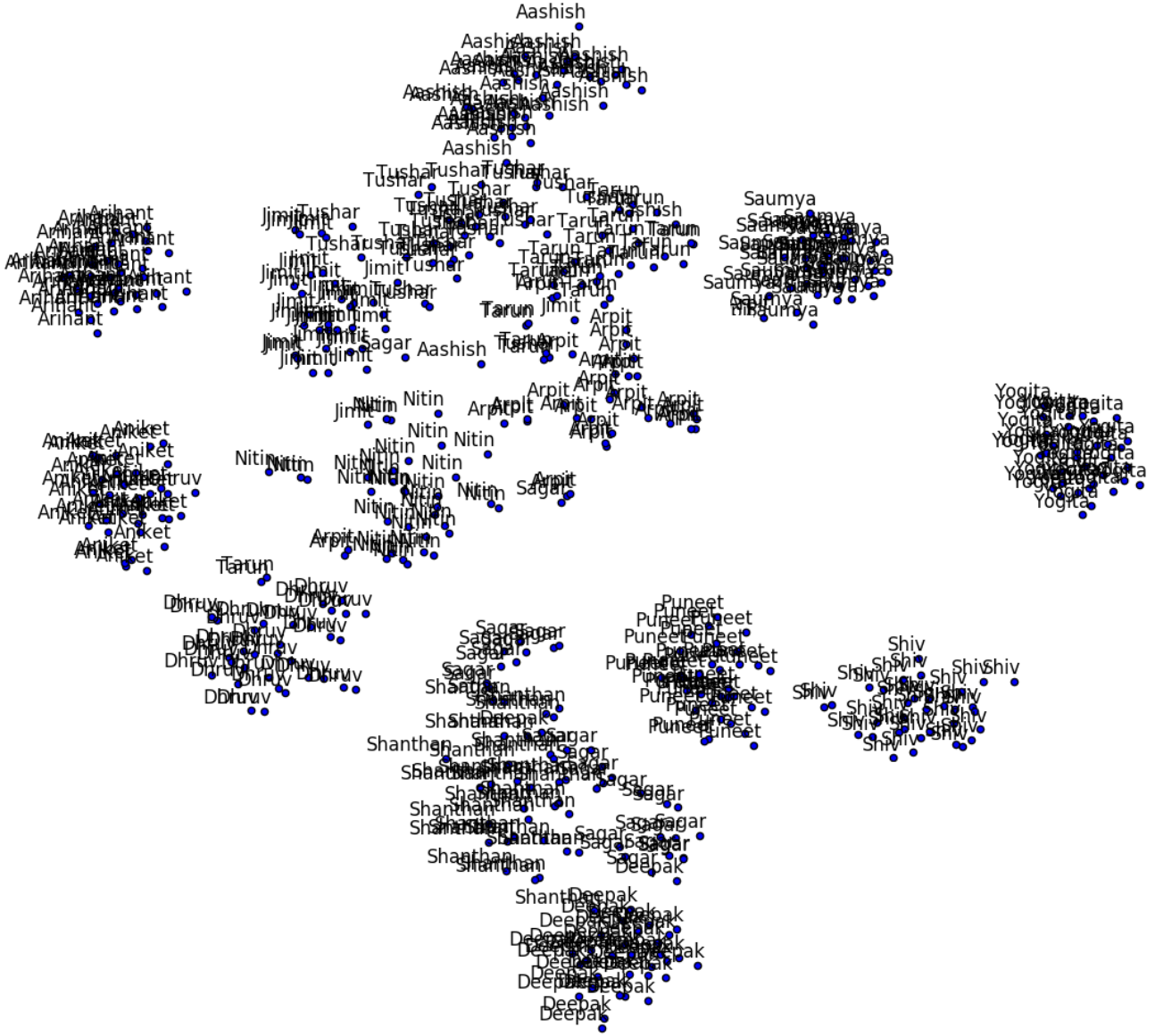
Dataset	No of Speaker	No. Of Examples
TIMIT	630	6300
VoxForge	6122	86504
Total	**6860**	137061

Input Specs: MFCC + DELTA + DELTA-DELTA Features	
Window Length	25ms
filters	64
sample Freq	16 KHz
Delta(acceleration)	Used
Delta-Delta(velocity)	Used
Feature length	64
Input sample duration	~655ms

SPEAKERNET TRAINING DETAILS

- Convolution Neural Network is inspired by DeepSpeaker's conv+res neural network.
- Used ADAGRAD optimizer with exponential decay learning rate.
- Alpha margin set to 0.5 (Euclidean distance) to choose triplets.
- TIMIT test data used to validate the SpeakerNet performance.

SPEAKERNET EMBEDDING VISUALIZATION



SPEAKERNET WITH CLASSIFIER

- Different classifier tested for classification.

Classifier	Accuracy(Test Data)
LinearSvm	76.8
GridSearchSvm	79.8
RadialSvm	86.7
DecisionTree	69.5
GaussianNB	65.1
MLPClassifier	70
Nearest_Neighbors	84.4
Gaussian_Process	82.3
Random_Forest	75.4
AdaBoost	81.3
QDA	79.7

- Results mentioned above are based on local test dataset generated for classifier training; RadialSVM classifier performs best with 86.7% accuracy.

SELECTION

- Send SpeakerSure as True to arbiter if SpeakerID satisfies below condition:
  - Classification confidence >= 0.85
  - Euclidean Distance with same person enrolled data <= 1.0
  - Cosine similarity with same person enrolled data >= 0.85

SELECTION

- Send FaceSure as True to arbiter if FaceID satisfies below condition:
  - Classification confidence >= 0.75
  - Euclidean Distance with same person enrolled data <= 1.0
  - Cosine similarity with same person enrolled data >= 0.85

EVOLUTION

- Compute Euclidean Distance and cosine similarity to decide which sample to retain. (both FaceID and SpeakerID)
- Retain 10 distinct images of each label for FaceID.
- Retain 40 Distinct Speaker feature of each label for SpeakerID.
- Trigger retraining if new samples were added.

EVOLUTION RESULTS

- With Additive learning framework, confidence increases with time and diverse training data.
- Face and Voice recognition confidence for a person increases by ~20% with time, as arbiter module selects diverse data and retrains the classifier.
- With Additive learning framework, we are also able to achieve more accuracy on our test dataset from 86.7% to ~90%.

ARBITER PERFORMANCE

- Time taken by arbiter module to generate embeddings and train the classifier:
  - **Speaker**: ~40 Secs *[15 subjects, ~25 sec audio/subject]*
  - **Face**: ~30 Secs *[15 subjects, ~5 images/subject]*
  - Hardware configuration used:
    - X86 Ubuntu 16.04
    - 1 TitanX GPU

REFERENCES

- **[1]OpenFace**: https://cmusatyalab.github.io/openface/
- **[2]Deep Speaker**: an End-to-End Neural Speaker Embedding System, arXiv:1705.02304
- **[3]MFCC feature extraction**: http://python-speech-features.readthedocs.io/en/latest/
- **[4]TensorFlow**: https://www.tensorflow.org/
- **[5]Protobuf**: https://developers.google.com/protocol-buffers/
- **[6]Python Modules**:
  scipy, scikit-learn, opencv-python, h5py, matplotlib, Pillow, requests, psutil pyaudio, numpy, xgboost, scikits.talkbox, sklearn, python_speech_features pandas
- **[7]SOX**: http://sox.sourceforge.net/
- **[8]FFMPEG**: https://www.ffmpeg.org/
- **[9]TIMIT**: https://catalog.ldc.upenn.edu/ldc93s1
- **[10]VoxForge**: https://old.datahub.io/dataset/voxforge

“Evolution, of course, is not something that simply applies to life here on earth; it applies to the whole universe.”

John Polkinghorne

“It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change.”

Charles Darwin

Thank you!

Inserting video: Insert/Video/Video from File.
Insert video by browsing your directory and selecting OK.

File type that works best in PowerPoint is: .wmv

VIDEO FILE

DEMO: PLACEHOLDER *(INSERT PICTURE BEHIND GRAY BAR)*