Practical Design and Training of Edge Deployable VQA in a Natural Interaction Generative Edge Pipeline

Danilo Pau, Gloria Giorgetti, Pietro Firpo

System Research and Applications

March 25th 2026

EdgeAI San Diego 2026

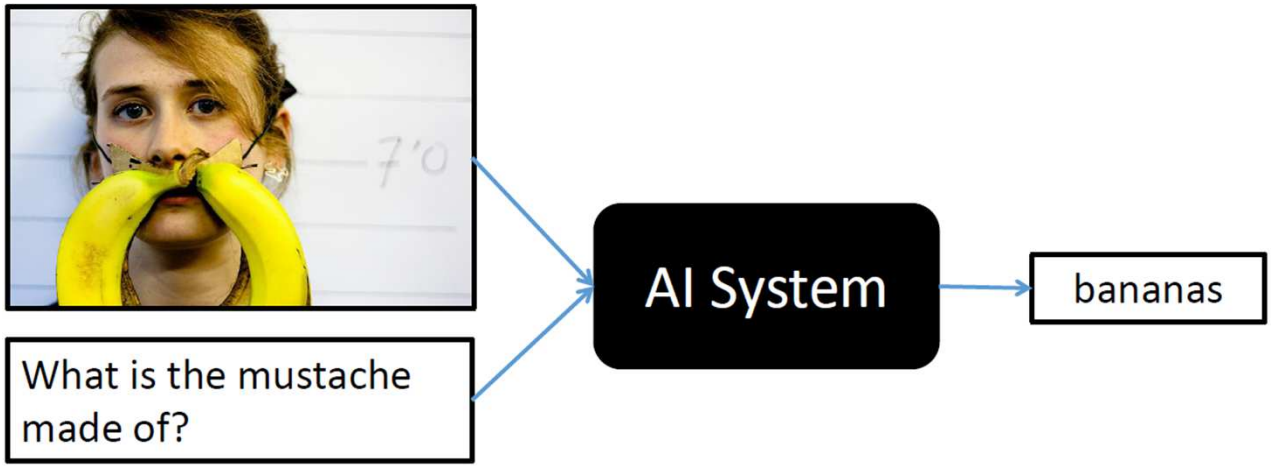
Agenda

- Introduction
- System Overview
- VQA Model Development
- Model Performance
- System Deployment
- Conclusions

Introduction

Visual Question Answering (VQA) for EdgeAI

- Input: image + text question → Output: predicted answer.
- Provides a **concrete case study and learn how to develop an advanced AI workload for an Edge processor**.



Approach and Contributions

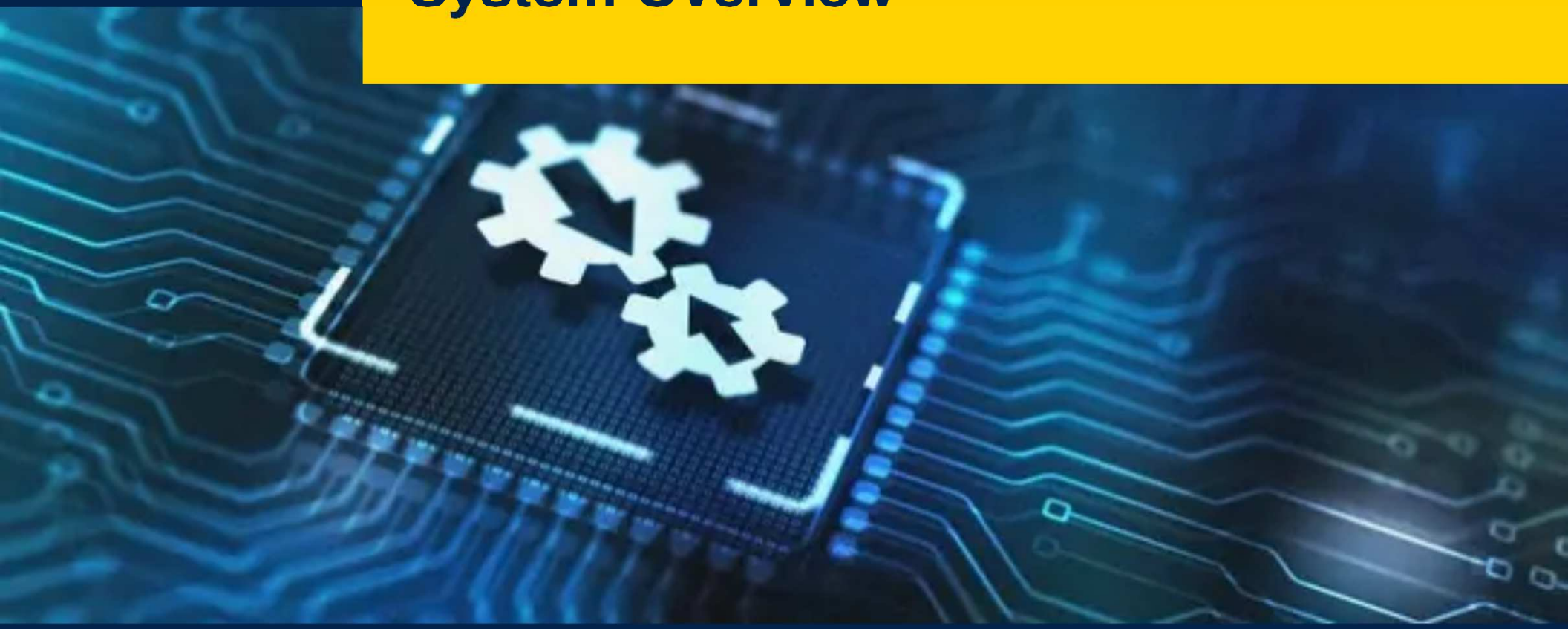
A Practical tutorial: building **lightweight VQA models** optimized for **STM32MP2** exploiting the embedded NPU.

- Practical, reproducible by everyone, designed for students and young professionals.

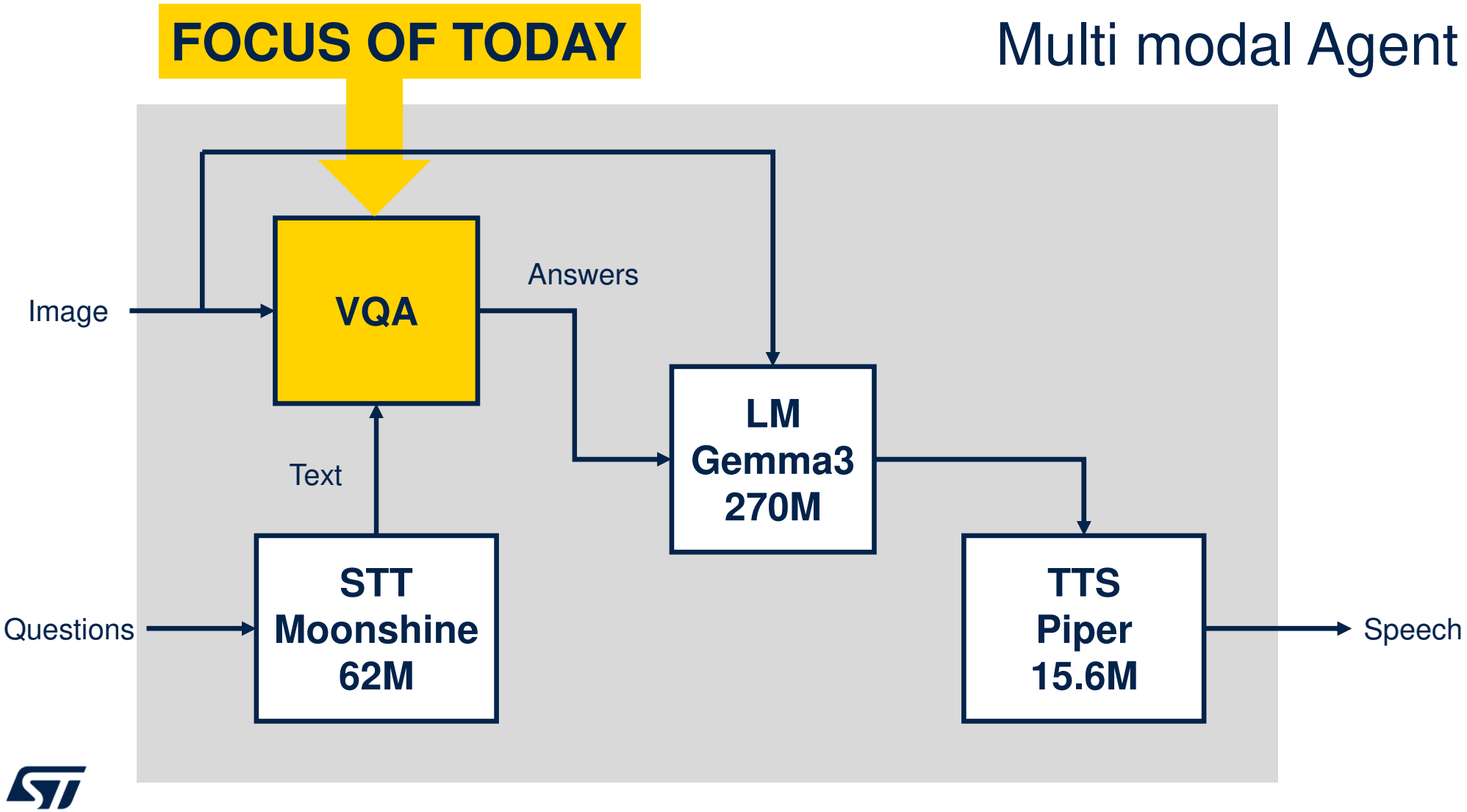
Help to build a modular generative edge system: given an image and a spoken question, generates a natural spoken answer directly on edge hardware

- Speech-to-Text (SST) - github
- **Lightweight VQA model - learn from this tutorial**
- Edge Language Model (ELM) – Foundational model
- Text-To-Speech (TTS) - github

System Overview



Multi modal Agent

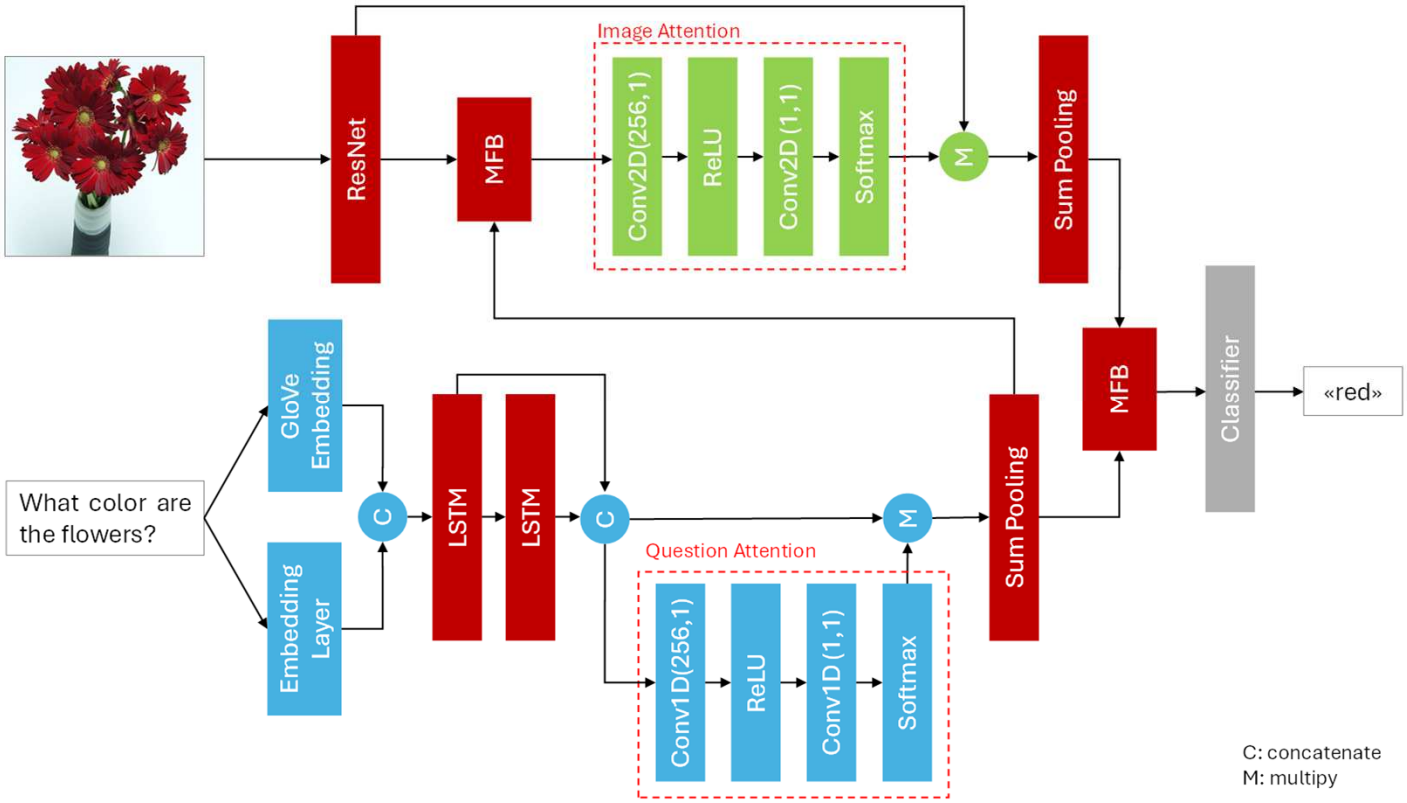


VQA Unfriendly Edge NPU Model

VQA Model Architectures

Architectures based on Yu et al. [1]:

- MFB **Baseline** (no attention layers)
- MFB **Attention** (only image attention layer)
- MFB **Co-Attention** (both image and question attention layers
- Feature extraction:
  - Text → LSTM
  - Image → ResNet-152



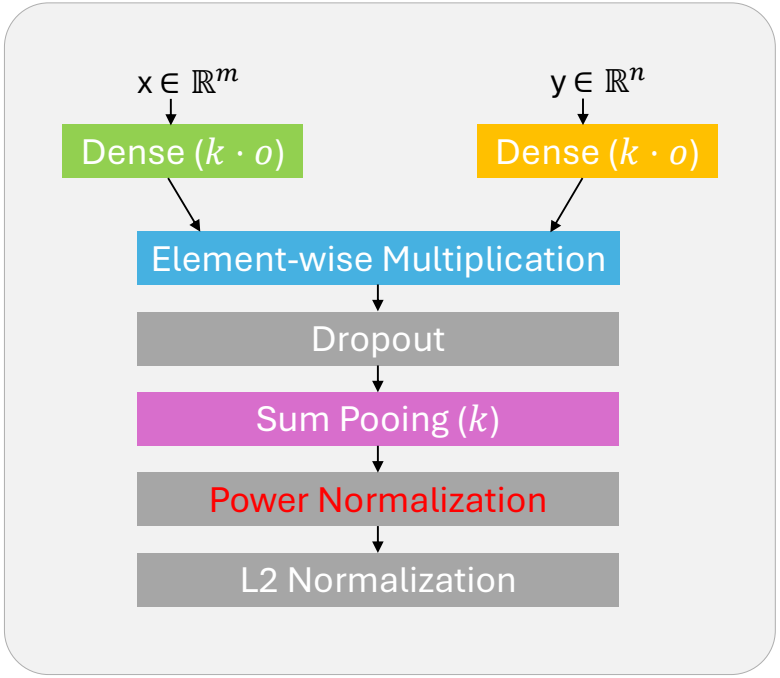
[1] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In Proceedings of the IEEE international conference on computer vision. 1821–1830.

VQA Model Architectures

Architectures based on Yu et al. [1]:

- Fusion: Multimodal Factorized Bilinear (MFB) module
- k and o are hyperparameters of the module

MFB module

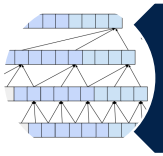


[1] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In Proceedings of the IEEE international conference on computer vision. 1821–1830.

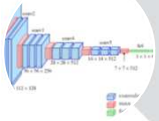
VQA Friendly Edge NPU Model

Optimizations STM32MP2 friendly

Optimizations aimed to achieve efficient execution with/without STM32MP2 NPU acceleration



LSTM replaced by Temporal Convolutions (TCLs)
For faster, hardware-friendly sequence modeling.



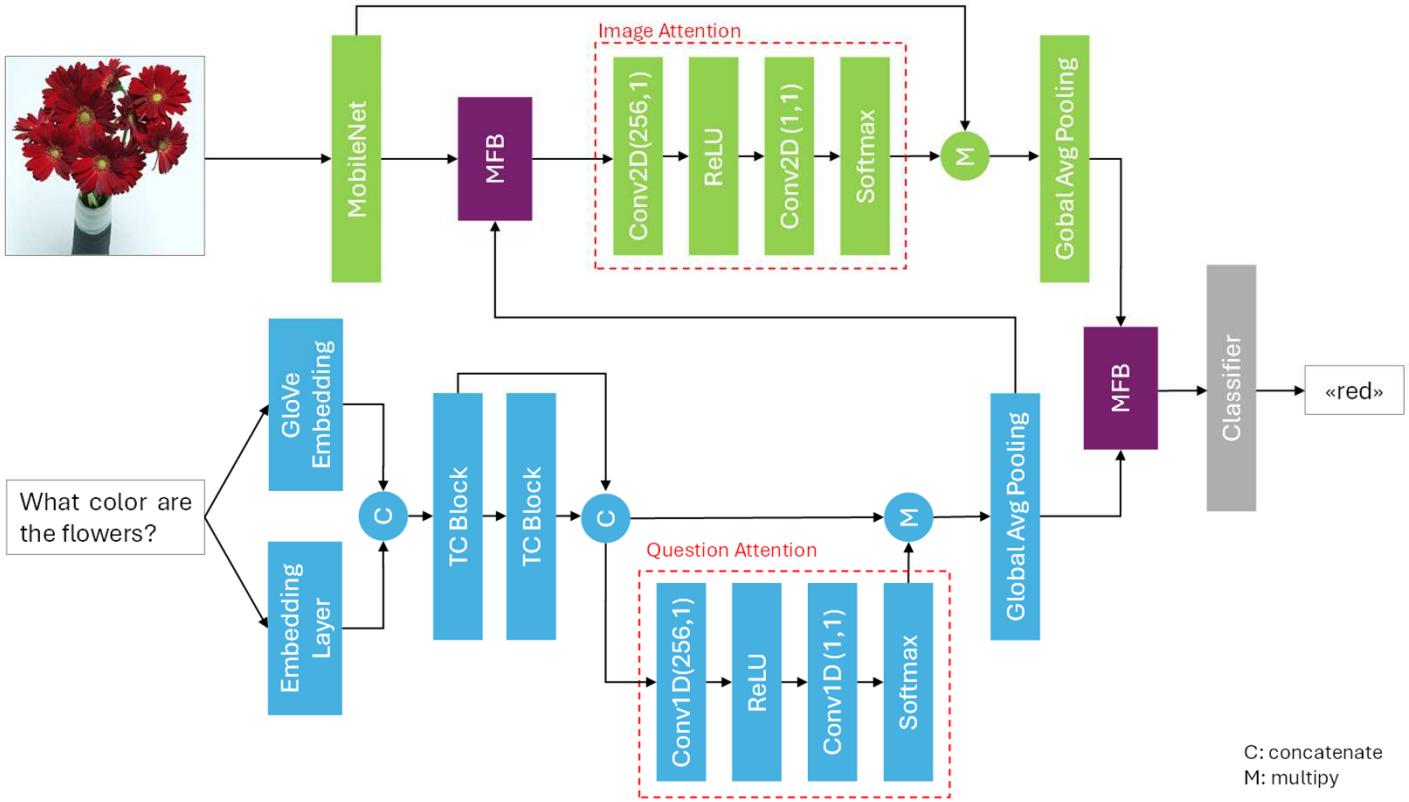
ResNet-152 (60.3M) replaced by MobileNetV3 Large (5.5M) more suitable for edge devices.

MFB simplifications: average pooling, removed power normalization

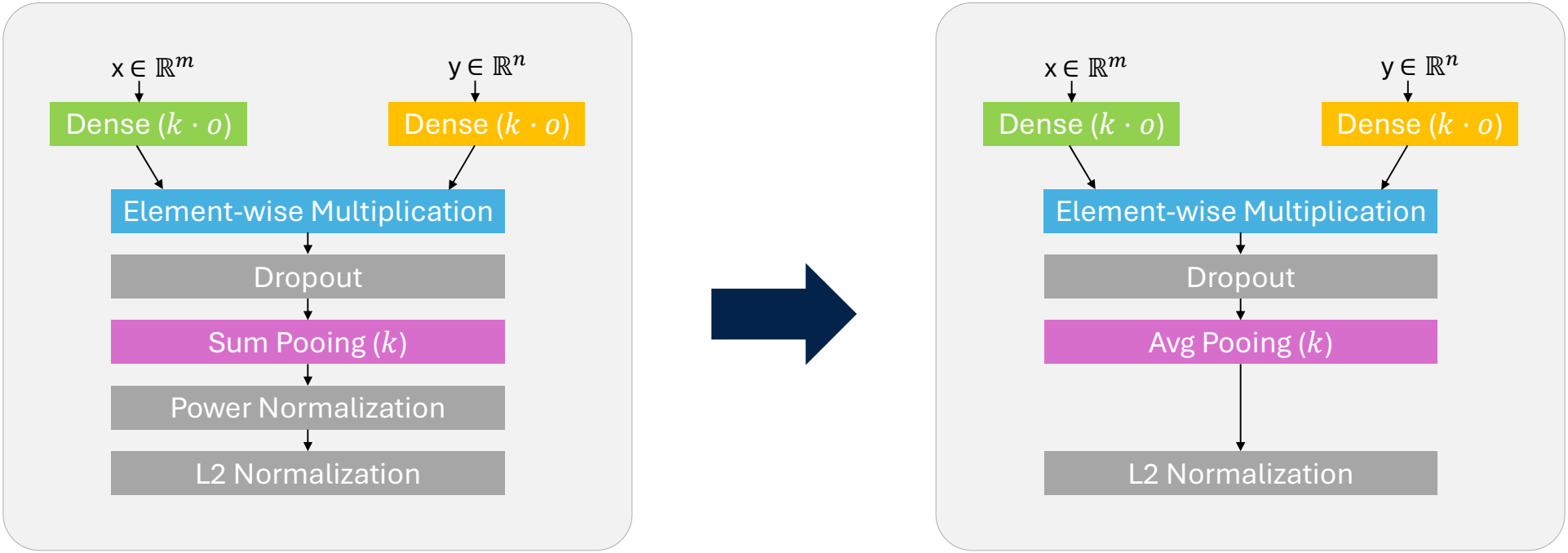
Word embeddings: concatenation of learned embeddings with pre-trained GloVe for richer semantics.

Optimized VQA Architectures

- **MFB CoAttention**

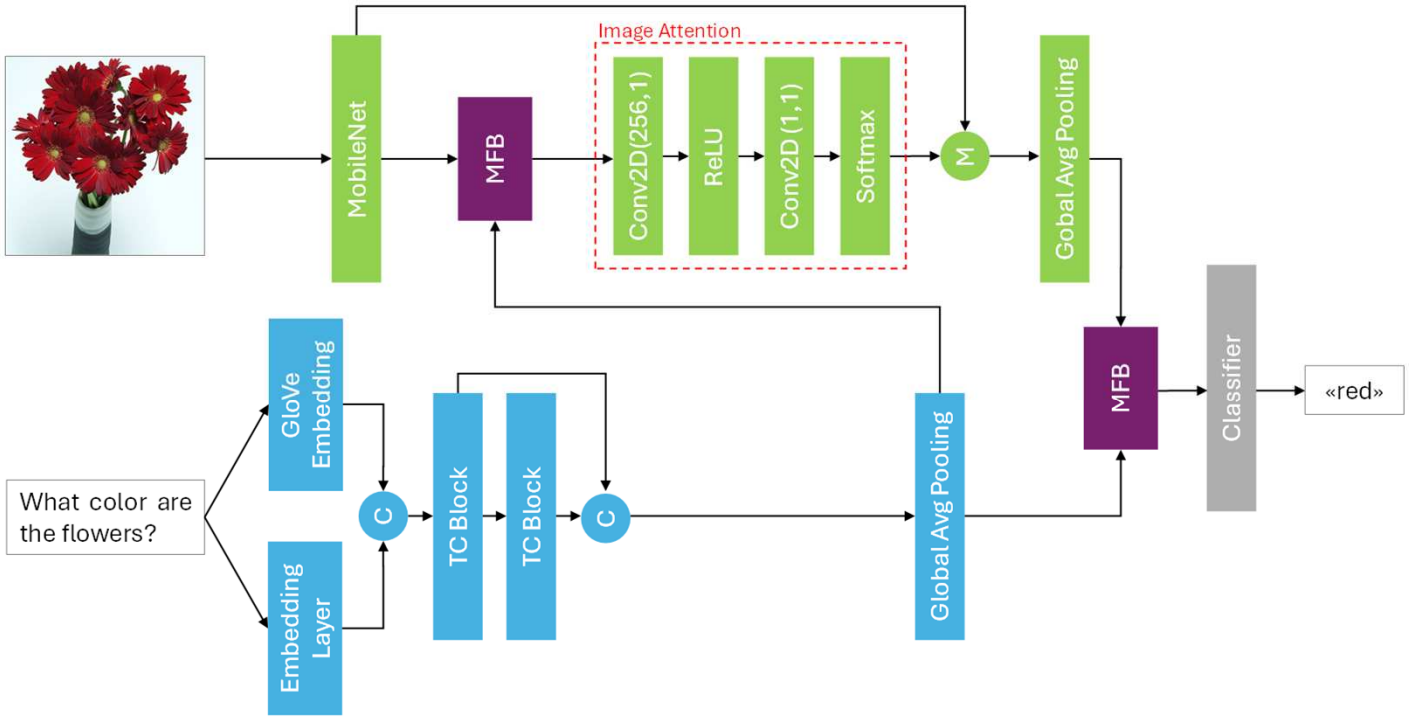


Optimized MFB



Optimized VQA Architectures

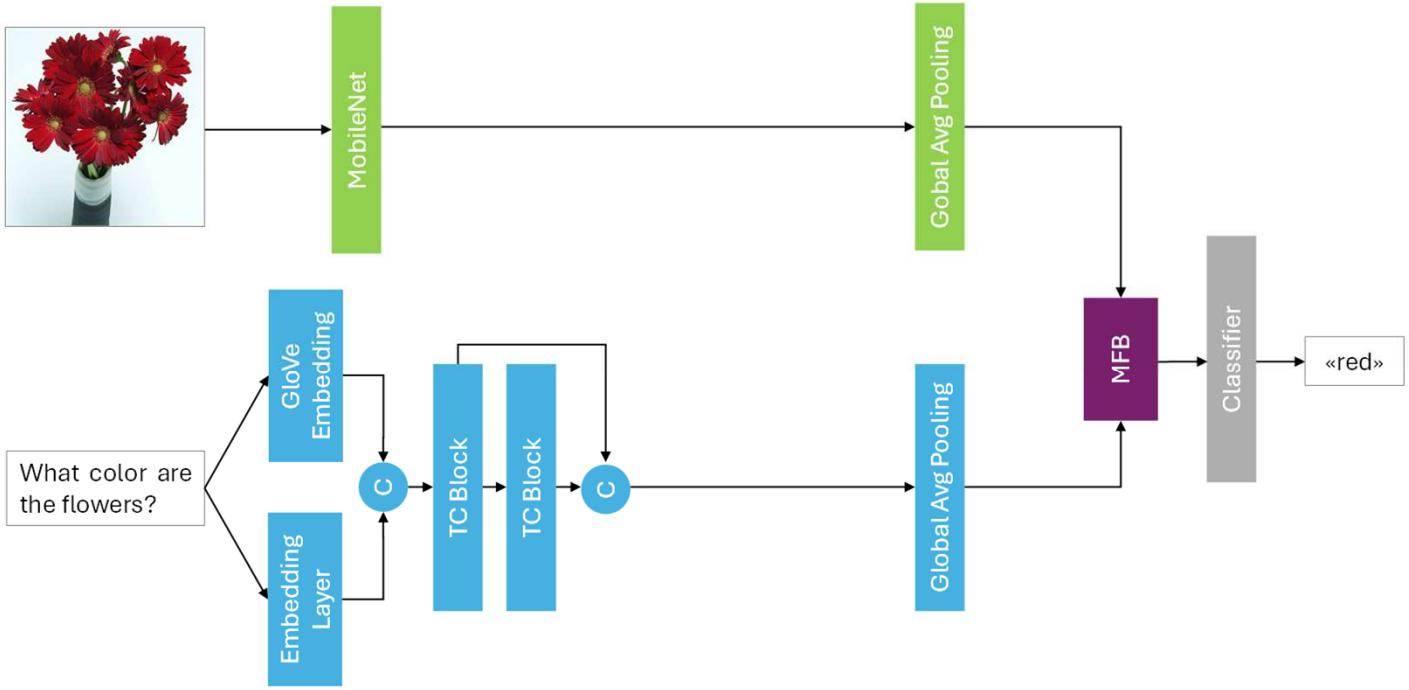
- **MFB Attention**



C: concatenate
M: multipy

Optimized VQA Architectures

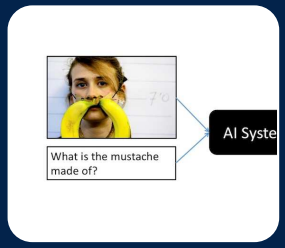
- **MFB Baseline**



Dataset

Dataset and Processing

Set	Questions	Images
Training	443,757	82,783
Validation	214.354	40,504



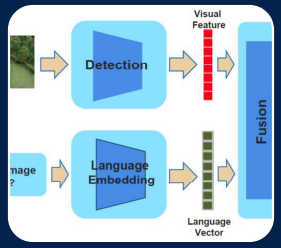
Dataset: VQA v2

~5 questions per image

1 ground truth answer + **10 human answers** per question

Pre-split into **train** / **val**

Evaluation performed on **validation set**

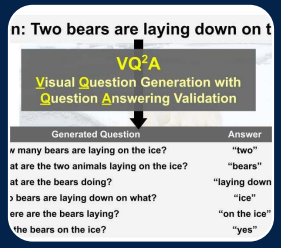


Processing:

Answers: normalized (official preprocessing guidelines)

Images: resized to 224x224, rescaled to [-1,1]

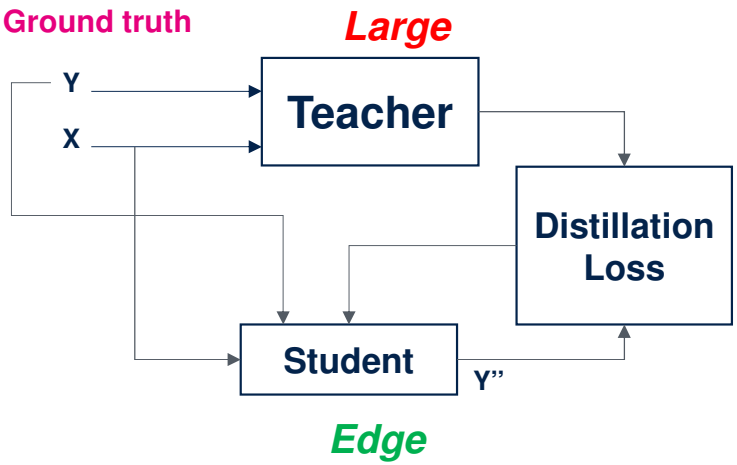
Questions: tokenized (vocabulary: 6415 tokens), padded/truncated to 15 tokens

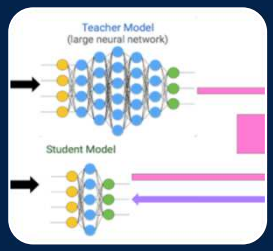


Answer space: top 1,000 most frequent answers

Training set reduced to 87.5% of original size

Training Procedure

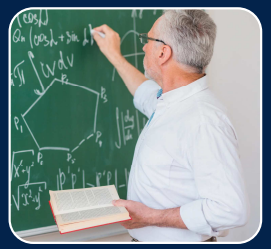




Knowledge Distillation (KD)

Student achieves higher accuracy than training from scratch

Method of Hinton et al. [2] (logit-based KD): combines cross-entropy (ground truth) and distillation loss (teacher's softened predictions)



Teacher model: BEiT-3 fine-tuned on VQAv2

Accuracy: **82.53%** (test-dev split)

3129 possible answers

683M parameters

Improvements:

Balanced loss to handle dataset imbalance and prevent overfitting to the most frequent answers

[2] Geoffrey Hinton, Oriol Vinyals, Jeff Dean
Distilling the Knowledge in a Neural Network,
20151503.02531, arXiv, https://arxiv.org/abs/1503.02531

Model Performance

Model Performance

- **Models vs. Original MFB and BEiT-3:**
  - Models achieve lower accuracy than original MFB model (65.4%) and SOTA
  - Also lower accuracy than teacher model BEiT-3 (87.5%)
  - Expected due to basic processing and simplified training procedure
- **Trade-off:** fewer parameters and efficient on STM32MP2 but reduced accuracy
- **Best results: MFB att.** → integrated into the generative system and **benchmarked on STM32MP2 (NPU)**

Model	#Params	Accuracy
MFB (original)	68.0 M	65.4%
MFB base.	24.4 M	56.0%
MFB att.	**40.4 M**	**57.0%**
MFB co-att.	51.2 M	56.4%

Top-10 ans. (highest accuracy)	Ans. 11-50	Ans. 51-1000
92.1%	85.1%	63.1%

Average accuracy of **MFB att.**, computed by ranking the 1000 possible answers by their per-answer accuracy."

System Deployment

VQA Acceleration on STM32MP2

Benefits of NPU acceleration:
Reduced runtime (~13.56× speed-up)
Reduced power consumption

Models (this tutorial)	Execution Device	Inference time [ms]	NPU usage %
MFB att.	**GPU/NPU**	**17.8**	**96**
MFB att.	2 A35 CPUs	241.41	0

Use case analysis on STM32

- Runtime (RTF/inference time/throughput), average power consumption, memory footprint
- $RTF = \frac{processing\ time}{audio\ lenght}$

Model	Framework	Execution Device	Runtime Measure	Power (W)	Flash (MB)	RAM (MB)
Moonshine	**Onnx runtime**	CPU	Real time factor: 1.50	0.89	244.8	1101
MFB att. (tutorial)*	**This work**	GPU/NPU	Inference time: 17.8 ms	0.75	46.9	69.6
Gemma 3 270M	**Ollama**	CPU	Throughput (tok./s): 4.07	1.00	292	789
Piper	**Onnx runtime**	CPU	Real time factor: 1.84	0.81	61	356


CONCLUSION

Tutorial @Github

Visual Question Answering (VQA) on Edge Devices

1. Introduction

This repository provides a hands-on tutorial that shows you how to design, train, and evaluate **Visual Question Answering (VQA)** models for **resource-constrained edge devices**, specifically the **STM32MP2 MPU**.

VQA is a task at the intersection of computer vision and natural language processing: given an **image** and a **natural language question** about it, the model must generate the correct **answer**. VQA has many real-world applications, such as accessibility tools for visually impaired users and smart assistants that understand visual content. However, most state-of-the-art VQA models are **large and resource-hungry**, making them unsuitable for **edge devices**.

The models in this tutorial are implemented in Keras 3.9, optimized for TensorFlow Lite, and designed to leverage the STM32MP2 NPU for efficient inference.

This README is structured as follows:

Conclusions

Tutorial:

Guide to develop lightweight multimodal VQA models

Deployable and accelerated on STM32MP2

Proof-Of-Concept Generative System:

Combines VQA + LM + STT + TTS

Enables multimodal and natural interaction on edge devices

Evaluated for latency, energy and memory on STM32MP2

Future Directions:

Build agents on STM32MP2

VQA model

"What is the boy doing?"

"Writing"

Generative System

"What is the boy doing?"

"He is writing a summary of yesterday math lesson"

danilo.pau@st.com

Our technology starts with You

Find out more at www.st.com