Segmentación De Materiales A Partir De Imágenes RGB Usando Arquitecturas De Transformadores De Visión E Integración De Información Multiespectral

Autor: Fabian Perez

Director: Hoover Rueda-Chacón
Codirector: Brayan Esneider Monroy Chaparro

Escuela de Ingeniería de Sistemas
Universidad Industrial de Santander
Bucaramanga, Colombia

AGENDA

01 Introduction

02 Objectives

03 Proposed Method

04 Simulations

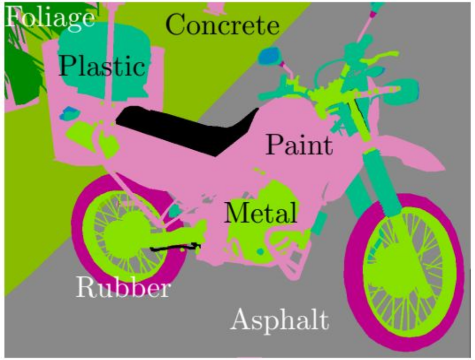
05 Results

06 Conclusions

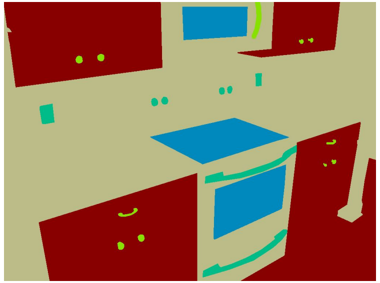
Introduction

Material Segmentation

Predict a dense segmentation map with material labels



Classes: {
Metal,
Plastic,
Asphalt,
...
}

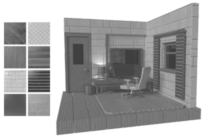


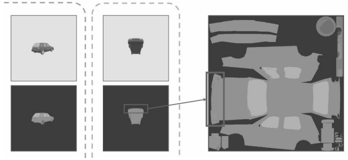
[1] Paul et al. A dense material segmentation dataset for indoor and outdoor scene parsing. ECCV 2022.

Applications of material segmentation

● Autonomous driving [2]

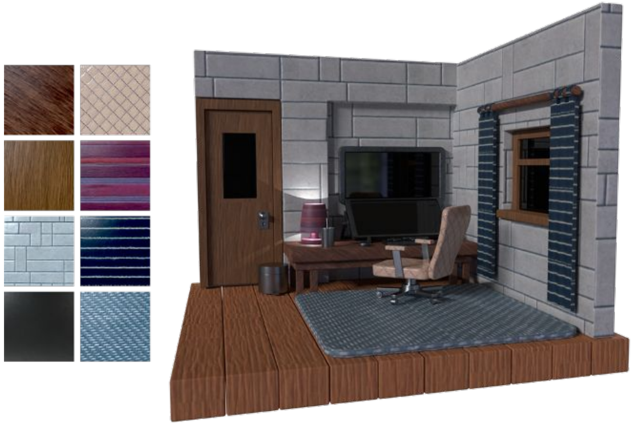
[2] Cai, S. et al. Rgb road scene material segmentation. ACCV 2022 (pp. 3051-3067)



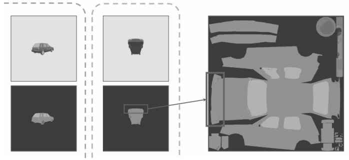


Applications of material segmentation

- Real-world simulation [3]



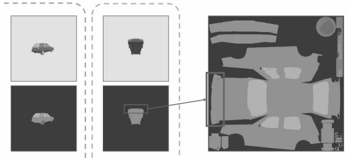
[3] Brandao, M. et al. Material recognition cnns and hierarchical planning for biped robot locomotion on slippery terrain. Humanoids 2016 (pp. 81-88).



Applications of material segmentation

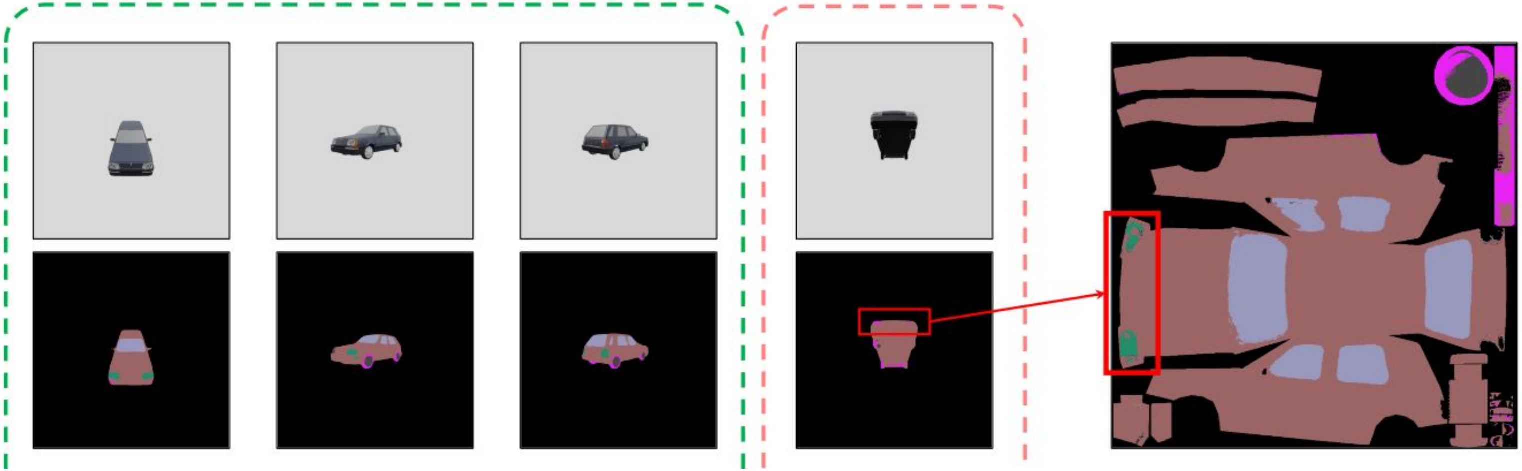
- Robot Navigation and decision [4]

[4] materobot: Material Recognition in Wearable Robotics for People with Visual Impairments

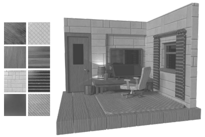


Applications of material segmentation

- 3D imaging [5]



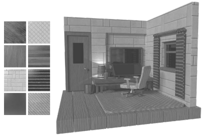
[5] Li, Zeyu, et al. "MaterialSeg3D: Segmenting Dense Materials from 2D Priors for 3D Assets." arXiv preprint arXiv:2404.13923 (2024).

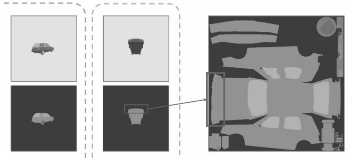


Applications of material segmentation

- Image Editing [6]

[6] Sharma, Prafull, et al. "Materialistic: Selecting similar materials in images." (2023).





Classical approach to material segmentation

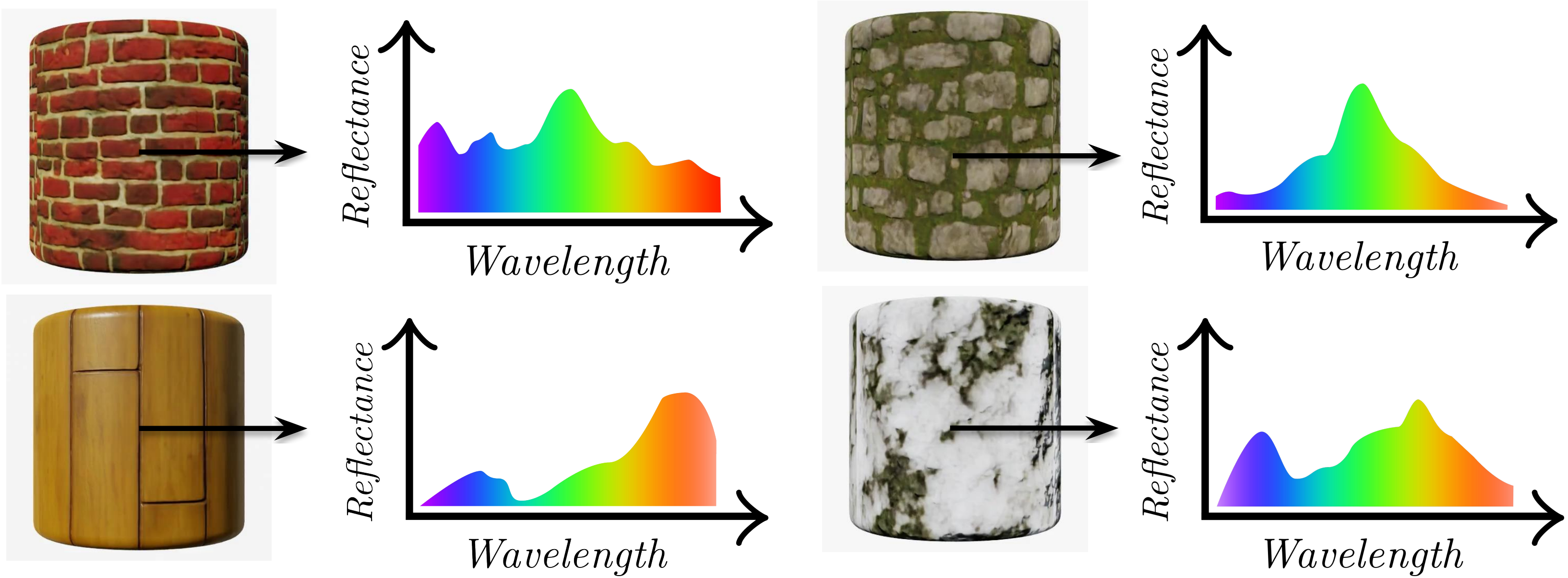
General approach: Segment only with RGB images which can be deceptive due to metamerism.

Chess patterns look the same but are of different materials [6]

Materials like plastic exhibit a wide range of appearances [6]

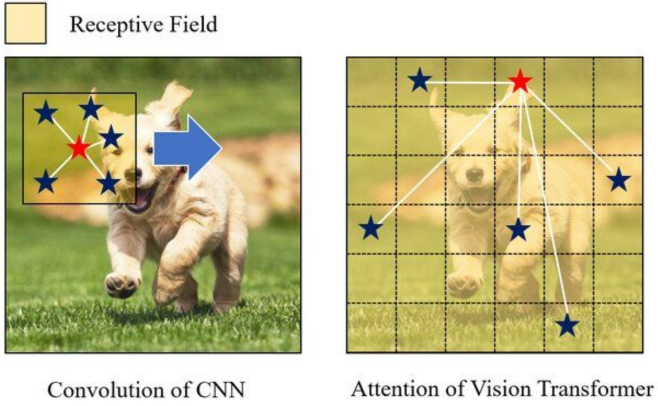
Spectral Information: A Key Component

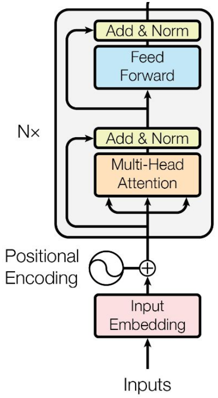
Each material exhibits a **representative spectral signature**, thus, enabling discrimination based on material properties



Vision Transformer

Image processing with attention mechanisms





[5] Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

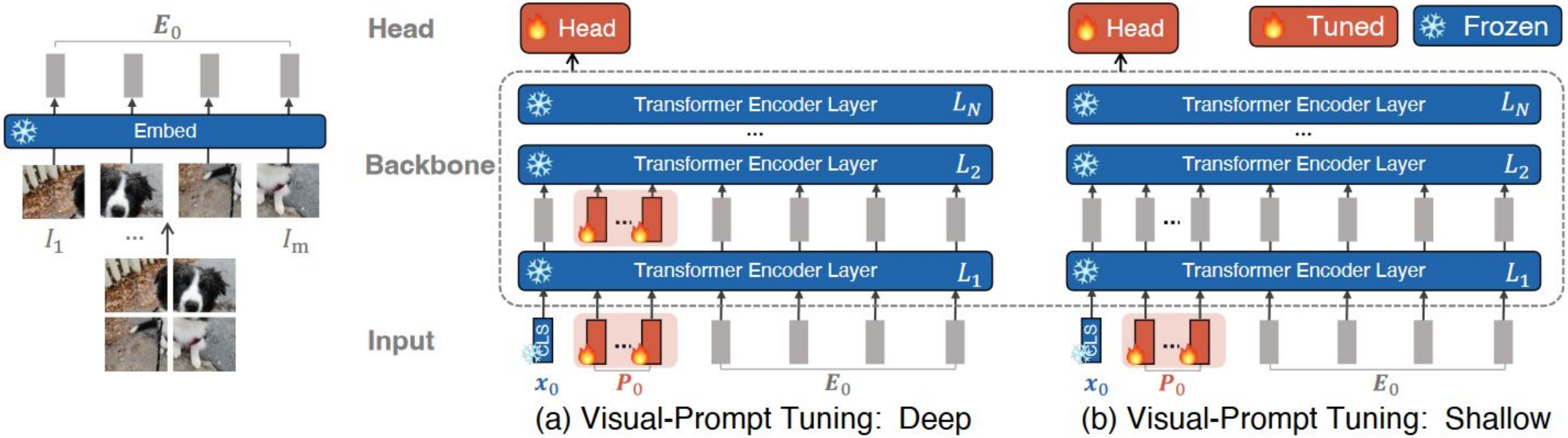
Vision Transformer

Anything you can tokenize, you can feed to Transformer

[6] LIMoE by B Mustafa, C Riquelme, J Puigcerver, R Jenatton, N Houlsby

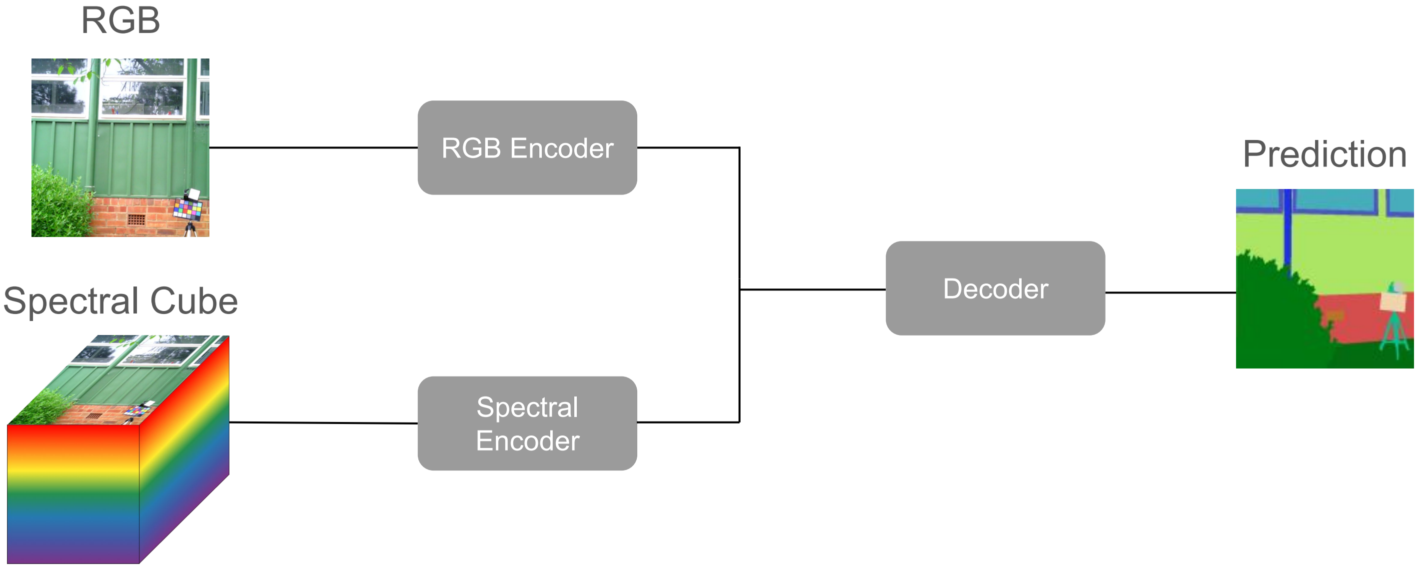
Prompt Tuning

Efficient adaptation through learnable task-specific tokens

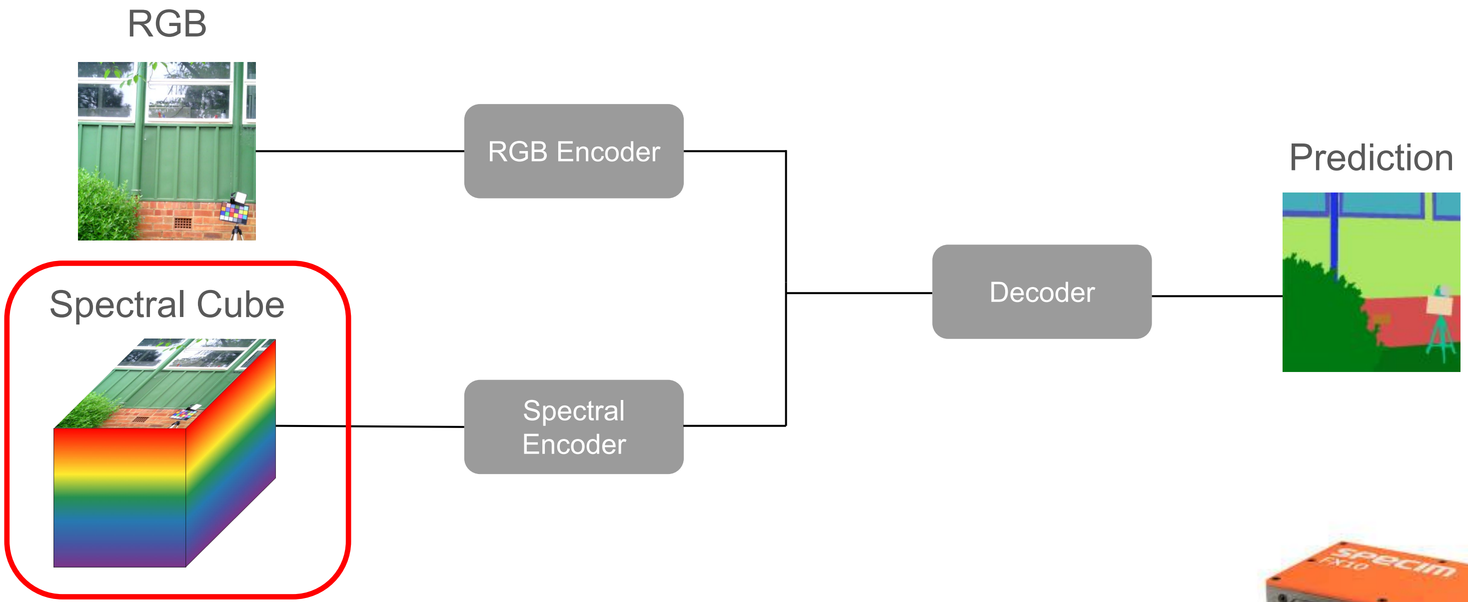


[7] Jia, Menglin, et al. "Visual prompt tuning." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.

Naive Approach for Material Segmentation

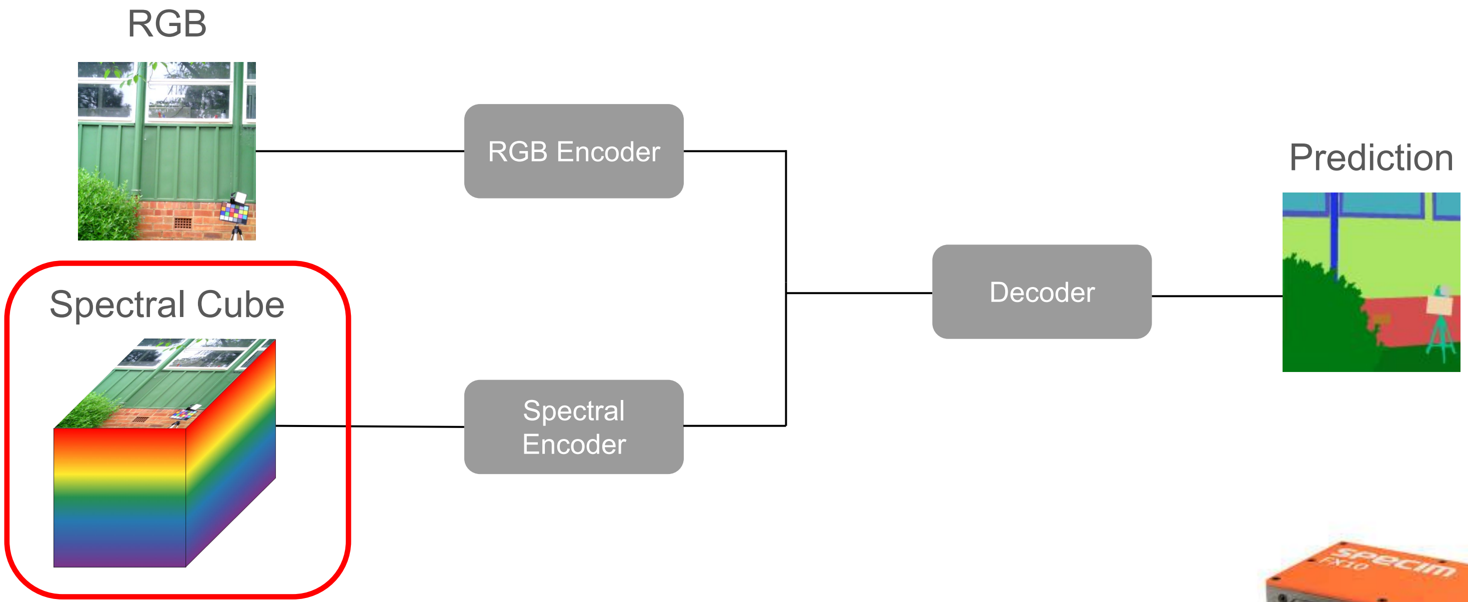


Naive Approach for Material Segmentation



- Significantly higher data acquisition and processing time
- Higher cost of implementation in practical systems
- Spectral sensors are not commonly available in consumer devices

Naive Approach for Material Segmentation



- Significantly higher data acquisition and processing time
- Higher cost of implementation in practical systems
- Spectral sensors are not commonly available in consumer devices

This approach requires the spectral cube at training and inference time.

Objectives

Objetivo General

Desarrollar y validar un algoritmo de segmentación de materiales basado en arquitecturas de Transformers de visión que integre información espectral sobre imágenes de color (RGB)

Objetivos Específicos

1. Identificar y seleccionar bases de datos de imágenes espectrales y de color (RGB) adecuadas para el entrenamiento y prueba del algoritmo

2. Diseñar una arquitectura de transformer de visión que integre información espectral y de color (RGB) de una escena para segmentarla en distintos materiales

3. Implementar en Python la arquitectura de transformer de visión diseñada para la segmentación de los materiales de una escena

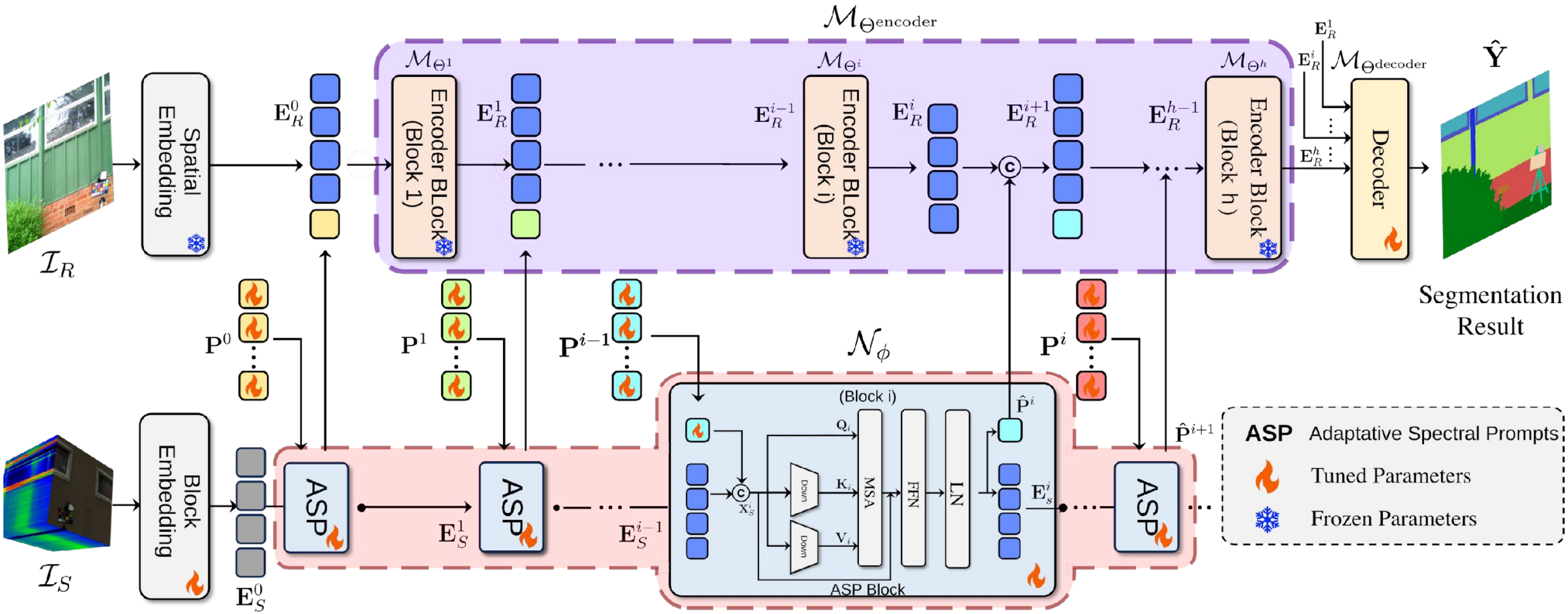
4. Evaluar el desempeño del algoritmo desarrollado mediante métricas de rendimiento estándar en el área de segmentación

5. Validar cualitativamente el algoritmo sobre un conjunto de imágenes de color adquiridas con una cámara disponible en dispositivos electrónicos de consumo

Proposed Method

Architecture overview

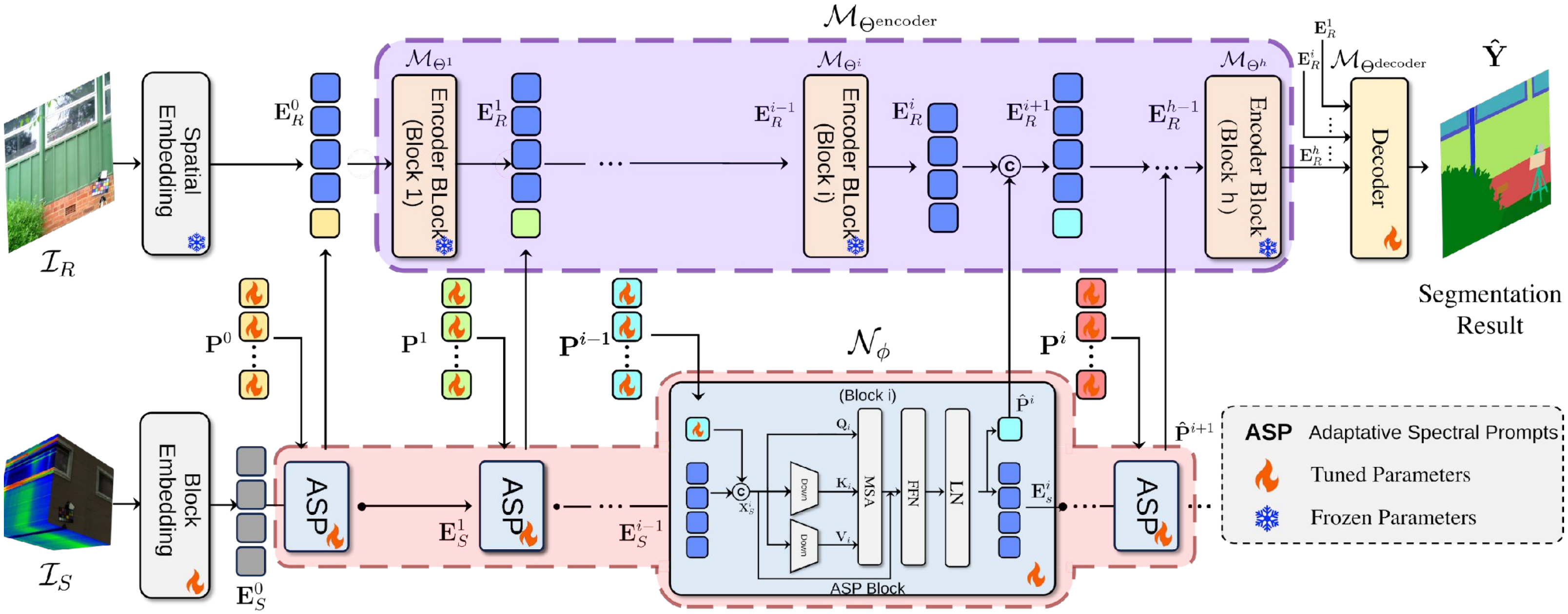
We propose a deep learning framework that bridges the gap between high-fidelity material segmentation and the practical constraints of data acquisition



Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H \times W \times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H \times W \times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H \times W}$

$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$

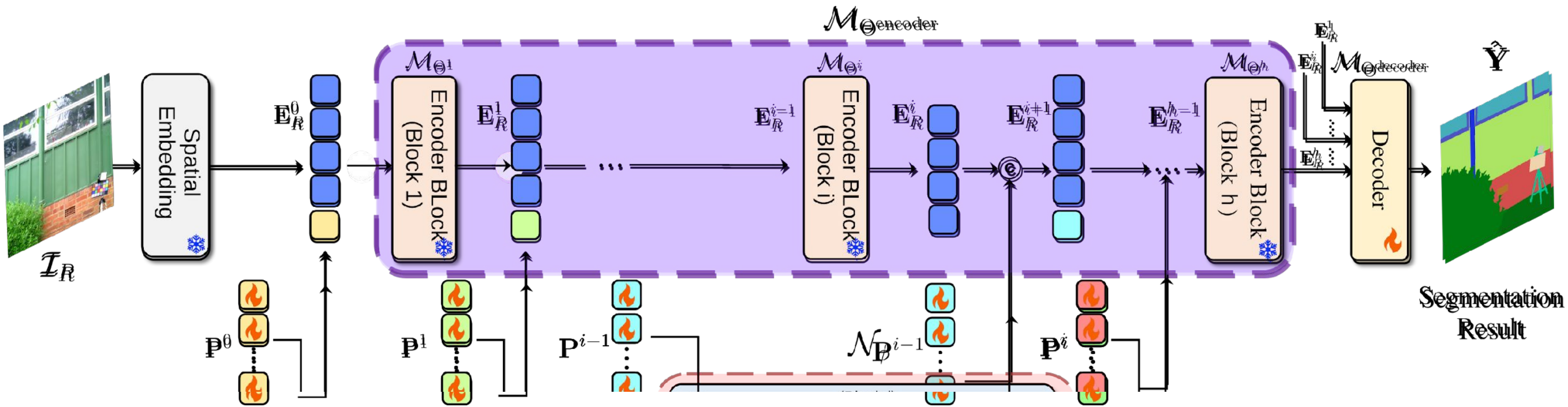


Architecture overview

RGB image: $\mathcal{I}_R \in \mathbb{R}^{H \times W \times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H \times W}$

If spectral image is not available:

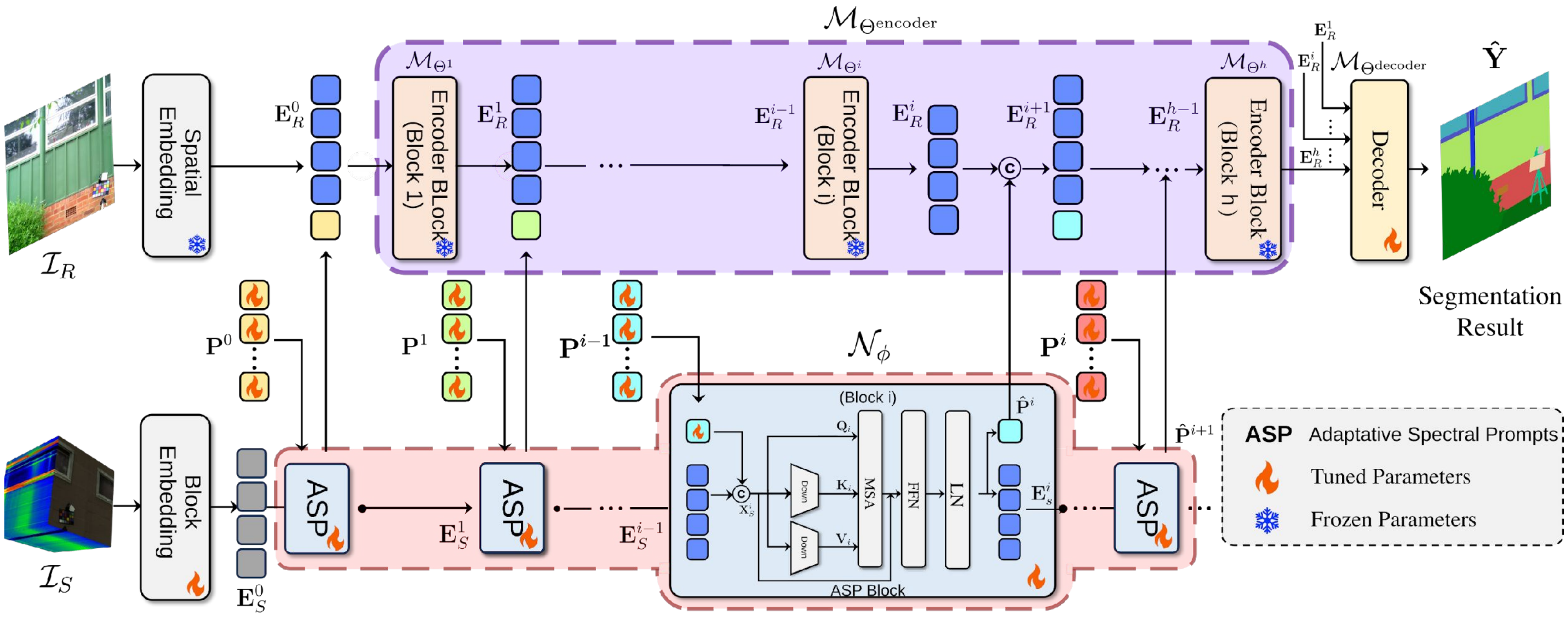
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{P}\right)$$



Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H\times W\times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H\times W\times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H\times W}$

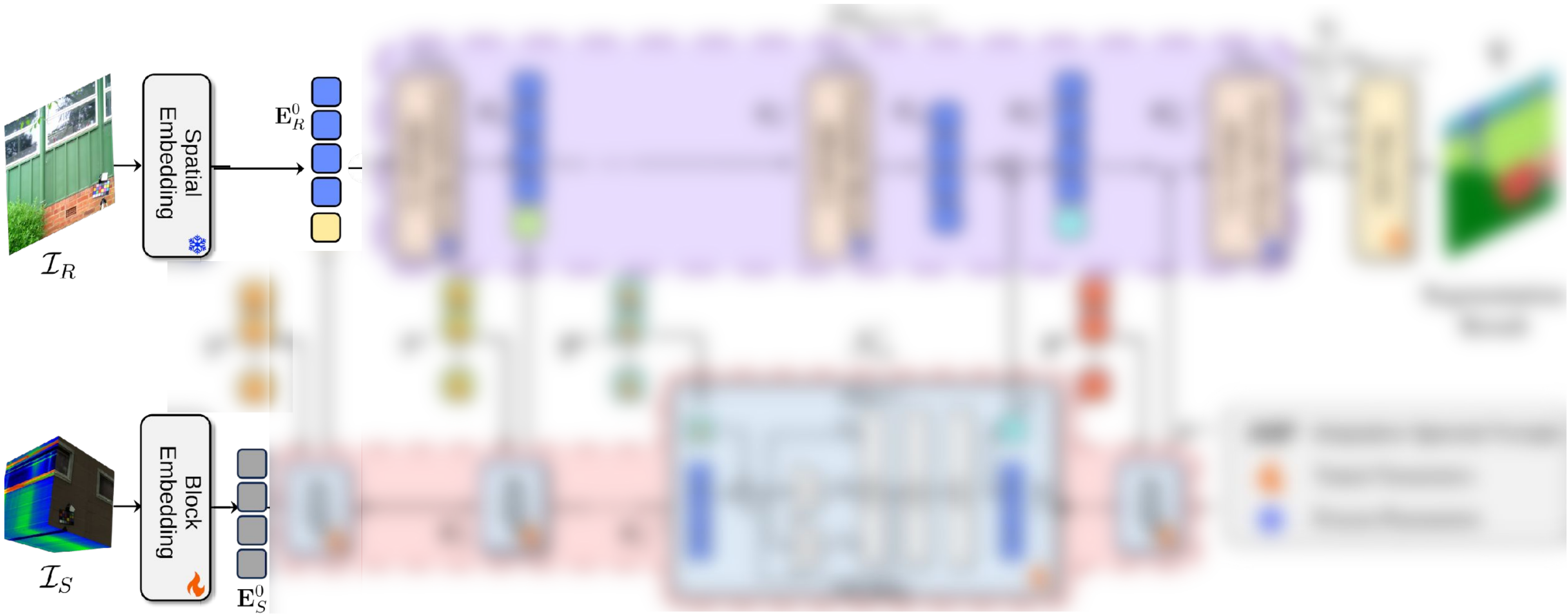
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



Architecture overview

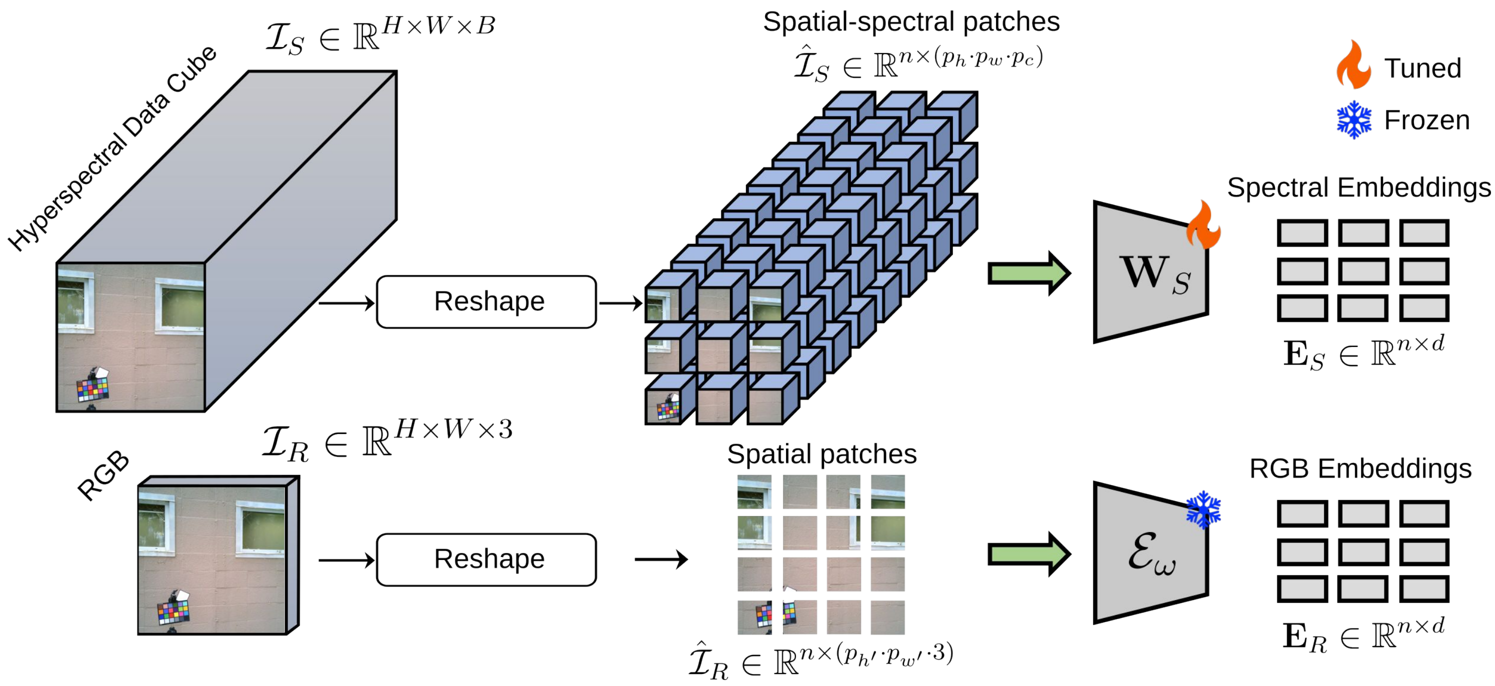
Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H\times W\times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H\times W\times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H\times W}$

$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



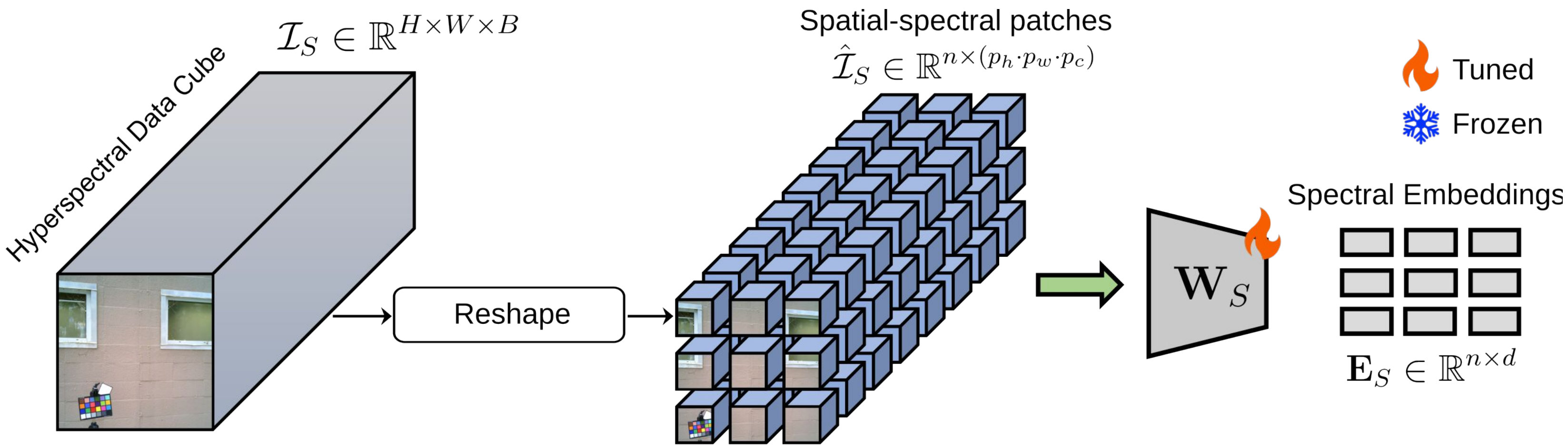
Embeddings

Given a dataset: $\mathcal{D} = (\mathcal{I}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$ we calculate the embeddings as follow:



Spectral Embeddings

Given a dataset: $\mathcal{D} = (\mathcal{I}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$ we calculate the spectral embeddings as follow:



Number of Patches

$$n = \left(\frac{H}{p_h}\right) \cdot \left(\frac{W}{p_w}\right) \cdot \left(\frac{B}{p_b}\right)$$

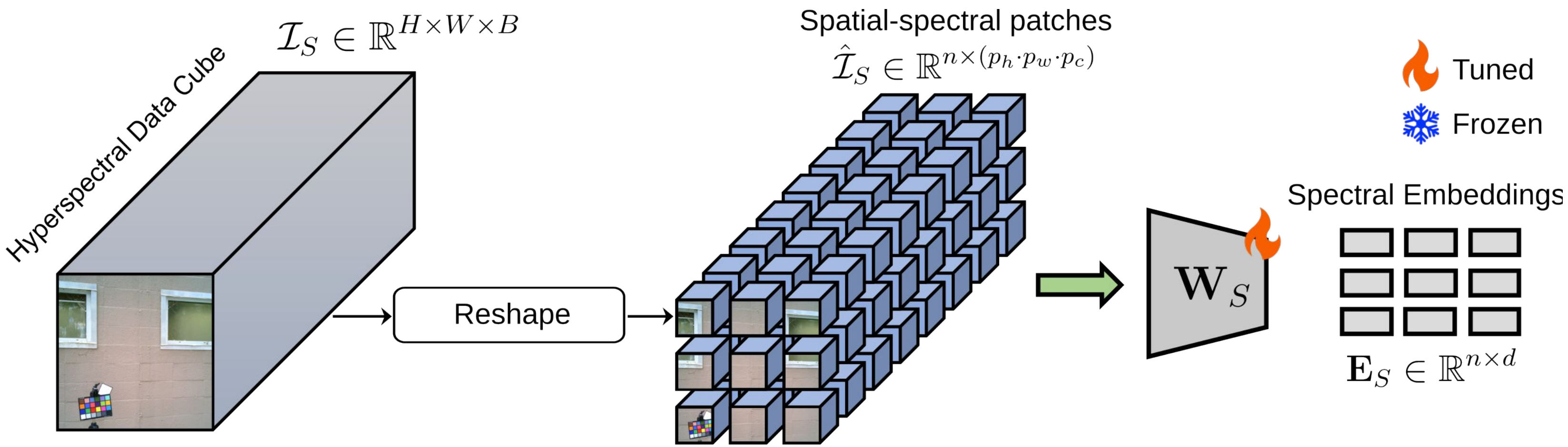
$$\mathbf{b}_k \in \mathbb{R}^{(p_h \cdot p_w \cdot p_c)}$$

Projection Matrix

$$\mathbf{W}_S \in \mathbb{R}^{(p_h \cdot p_w \cdot p_c) \times d}$$

Spectral Embeddings

Given a dataset: $\mathcal{D} = (\mathcal{I}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$ we calculate the spectral embeddings as follow:



Number of Patches

$$n = (\frac{H}{p_h}) \cdot (\frac{W}{p_w}) \cdot (\frac{B}{p_b})$$

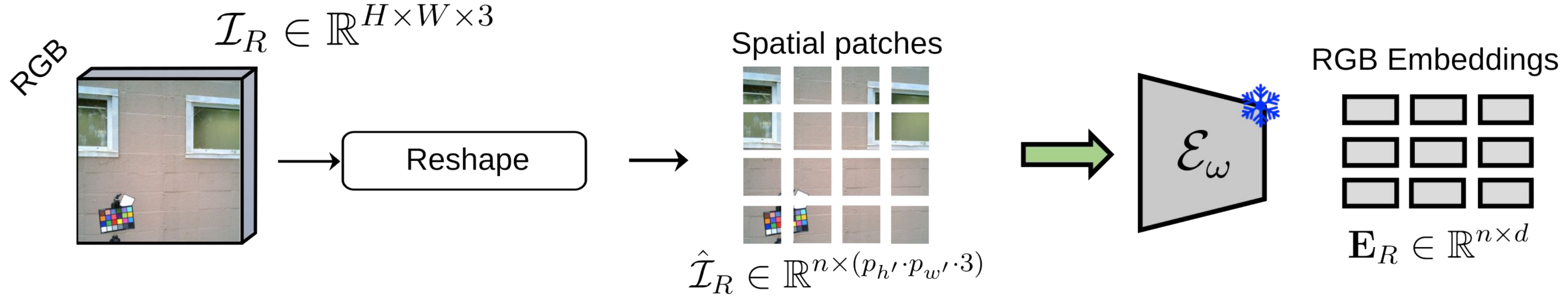
$$\mathbf{e}_k = \mathbf{W}_S^\top \cdot \mathbf{b}_k + \boldsymbol{\alpha}_k, \qquad \mathbf{e}_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, n.$$

Spectral Embeddings

$$\mathbf{E}_S = \{\mathbf{e}_k^\top \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq n\}$$

RGB Embeddings

Given a dataset: $\mathcal{D} = (\mathcal{I}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$ we calculate the RGB embeddings as follow:



Number of Patches

$$n = \left(\frac{H}{p_{H'}}\right) \cdot \left(\frac{W}{p_{W'}}\right)$$

$\mathbf{p}_k \in \mathbb{R}^{(p_{h'} \cdot p_{w'} \cdot 3)}$

$$\hat{\mathbf{e}}_k = \mathcal{E}_\omega(\mathbf{p}_k), \qquad \mathbf{e}_k \in \mathbb{R}^d, \quad k = 1, 2, \ldots, n$$

Embedding

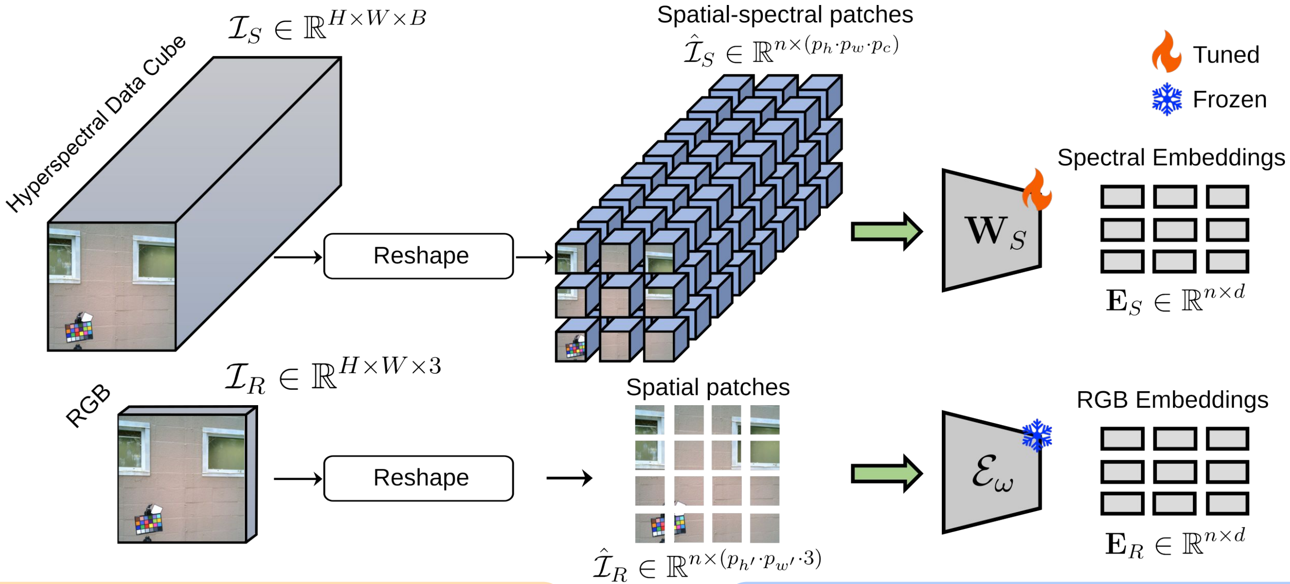
$$\mathcal{E}_\omega(\cdot)$$

RGB Embeddings

$$\mathbf{E}_R = \{\hat{\mathbf{e}}_k^\top \in \mathbb{R}^{d_i} \mid k \in \mathbb{N}, 1 \leq k \leq n\}$$

RGB Embeddings

Given a dataset: $\mathcal{D} = (\mathcal{I}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$ we calculate the RGB embeddings as follow:



Spectral Embeddings

$$\mathbf{E}_S = \{\mathbf{e}_k^\top \in \mathbb{R}^d \mid k \in \mathbb{N}, 1 \leq k \leq n\}$$

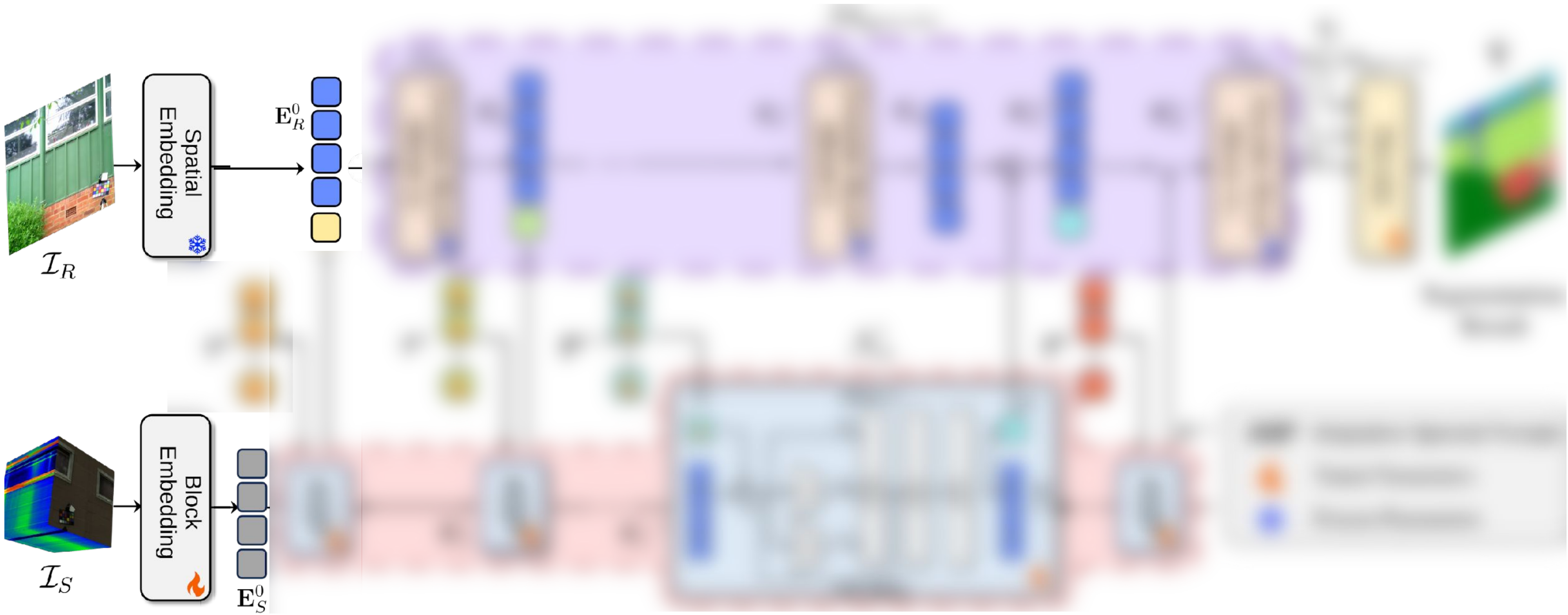
RGB Embeddings

$$\mathbf{E}_R = \{\hat{\mathbf{e}}_k^\top \in \mathbb{R}^{d_i} \mid k \in \mathbb{N}, 1 \leq k \leq n\}$$

Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H\times W\times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H\times W\times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H\times W}$

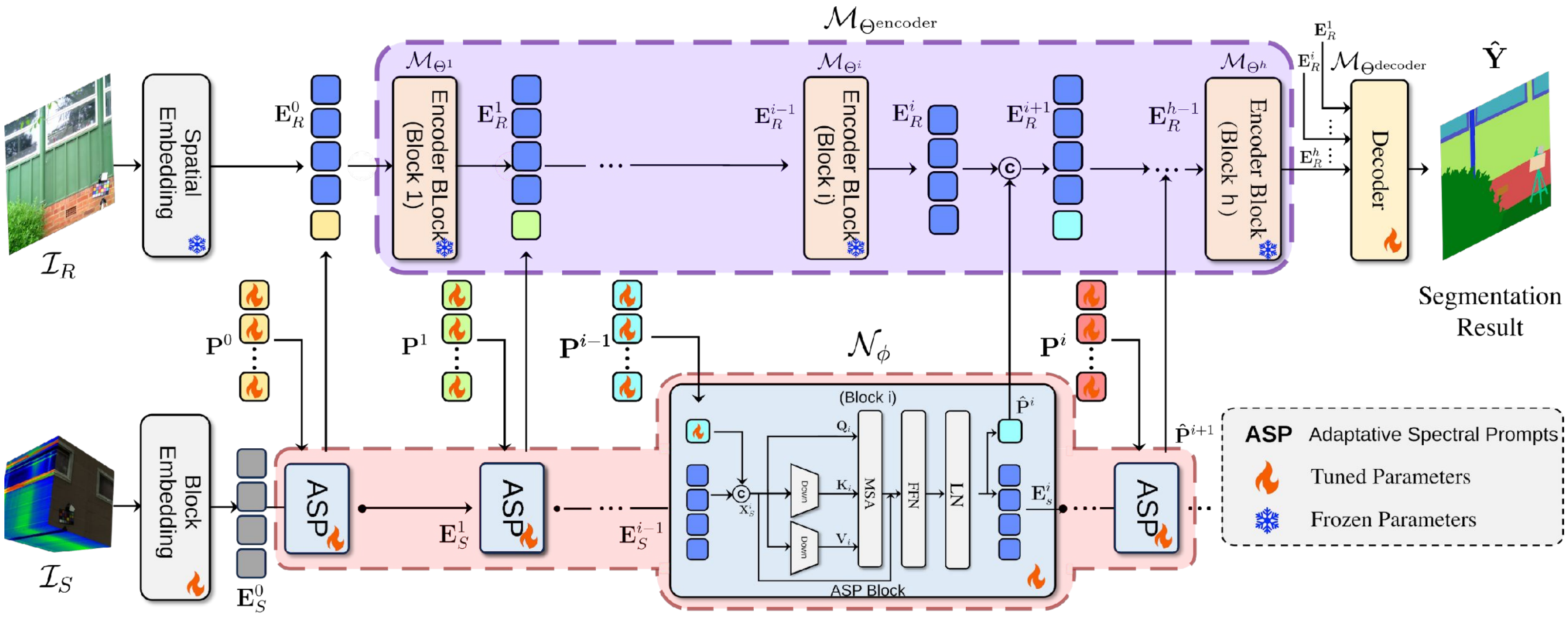
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H\times W\times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H\times W\times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H\times W}$

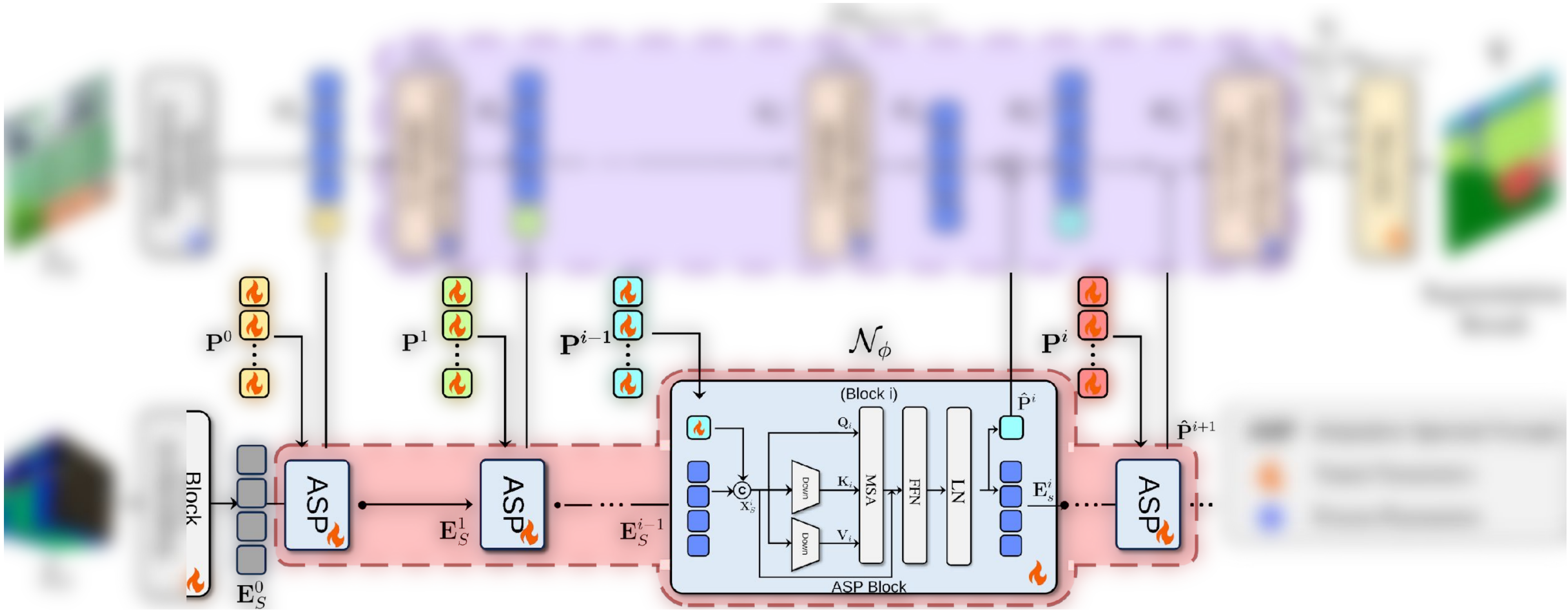
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H \times W \times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H \times W \times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H \times W}$

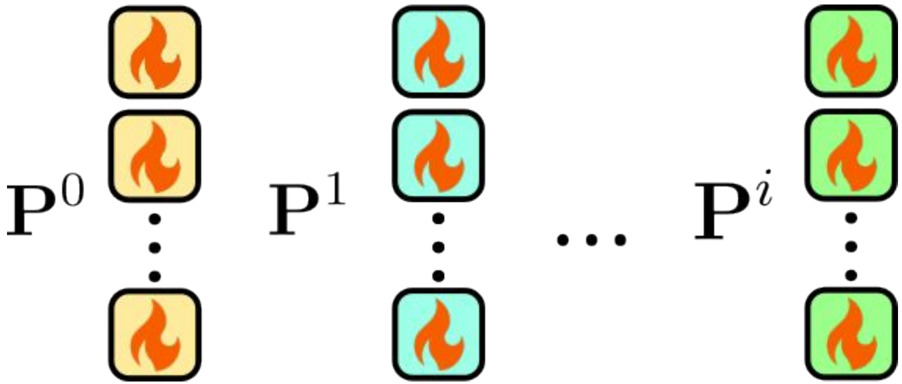
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



Adaptive Spectral Prompts (ASP)

We define a set of learnable prompts for each **block i** in the model

$$\mathcal{P} = \{\mathbf{P}^0, \mathbf{P}^1, \ldots, \mathbf{P}^h\}$$



Prompts at block i

$$\mathbf{P}^i = \{\mathbf{p}_l^\top \in \mathbb{R}^{d_i} \mid l \in \mathbb{N}, 1 \leq l \leq l_i\}$$

Adaptive Spectral Prompts (ASP)

We propose ASP to adapt spectral prompts with spectral information

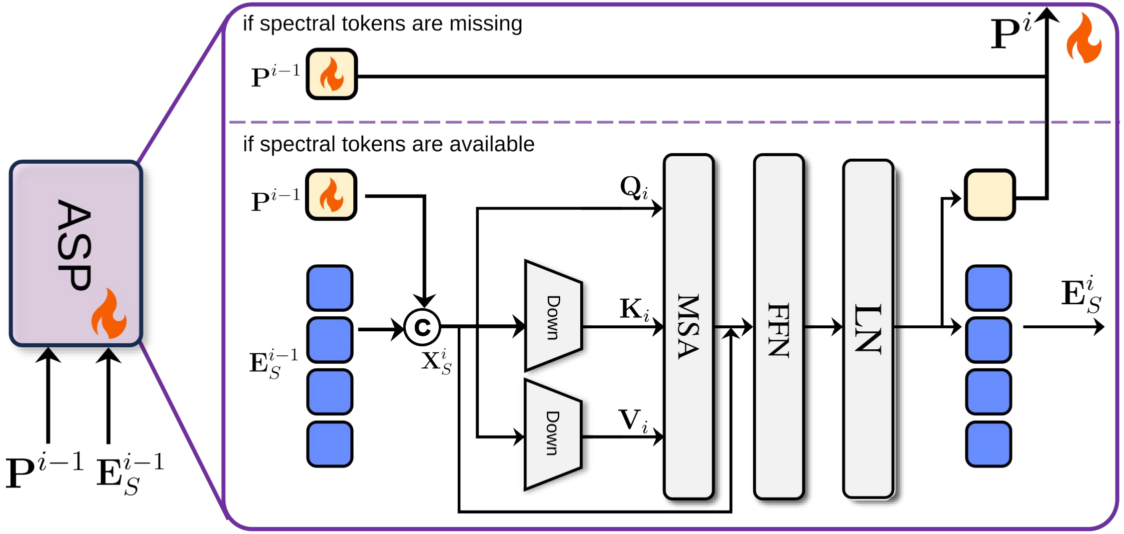
Prompts at block i-1

$$\mathbf{P}^{i-1} \in \mathbb{R}^{l_{i-1} \times d_{i-1}}$$

Spectral Embeddings

$$\mathbf{E}_s^{i-1} \in \mathbb{R}^{n \times d_{i-1}}$$

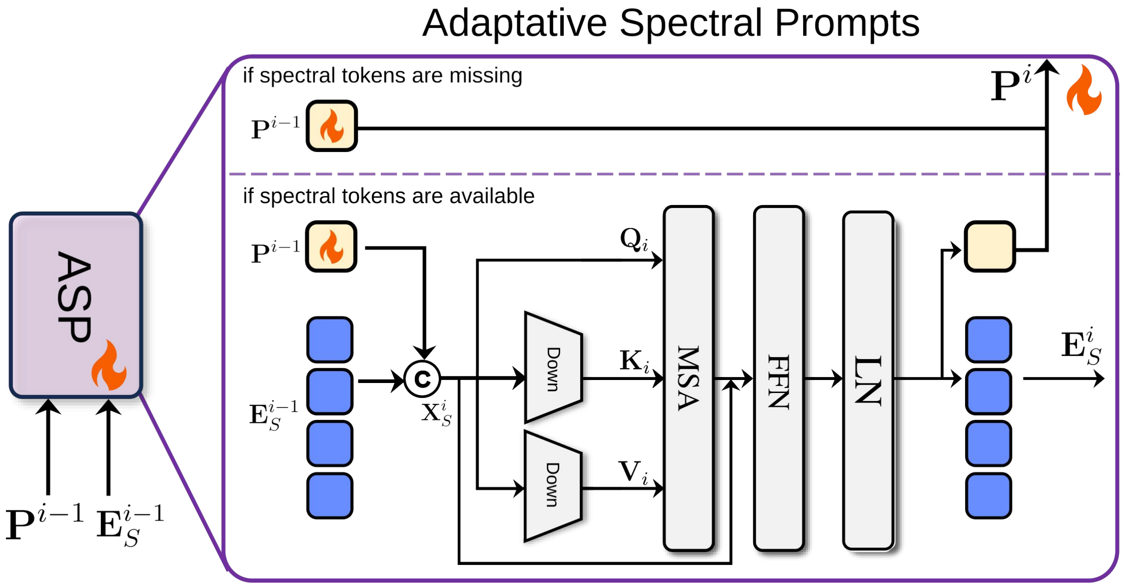
Adaptative Spectral Prompts



$$\mathbf{X}_S^{i} = [\mathbf{P}^{i-1}; \mathbf{E}_S^{i-1}] \in \mathbb{R}^{(l_{i-1}+n) \times d_{i-1}}$$

Adaptive Spectral Prompts (ASP)

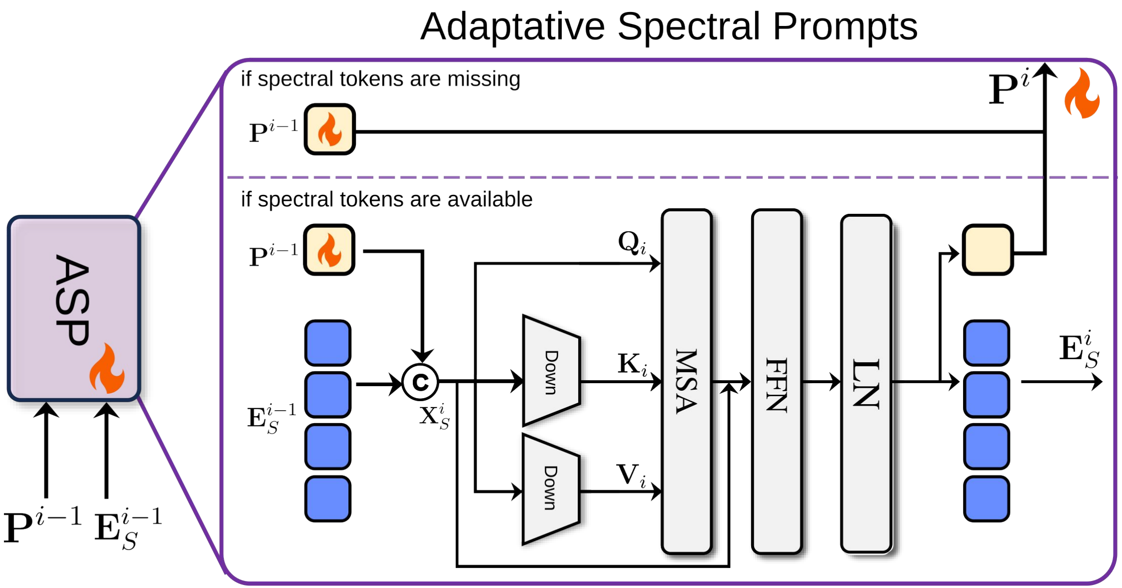
We propose ASP to adapt spectral prompts with spectral information



$$[\hat{\mathbf{P}}^i, \mathbf{E}_S^i] = \mathrm{ASP}_i(\mathbf{X}_S^i) = \mathrm{ASP}_i([\mathbf{P}^{i-1}; \mathbf{E}_S^{i-1}])$$

Adaptive Spectral Prompts (ASP)

We propose ASP to adapt spectral prompts with spectral information



$$[\hat{\mathbf{P}}^i, \mathbf{E}_s^i] = \mathrm{LN}^i(\mathrm{FFN}_\delta^i(\mathrm{MSA}_\psi^i(\mathbf{X}_S^i) + \mathbf{X}_S^i))$$

Multi-head self-attention

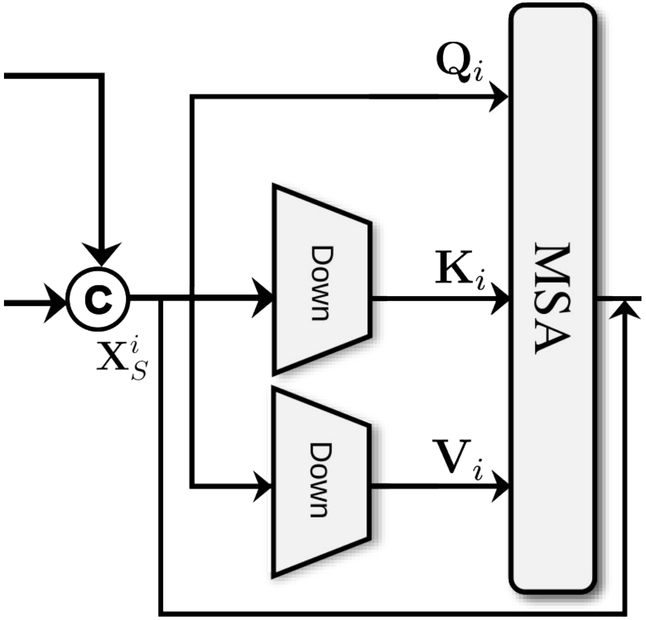
$$\begin{aligned} \mathcal{A}^i &= \mathrm{MSA}_\psi(\mathbf{X}_S^i) = \mathrm{Concat}(\mathrm{head}^1, \ldots, \mathrm{head}^c)\mathbf{W}^O, \\ \text{con} \quad \mathrm{head}^t &= \mathrm{Attention}(\mathbf{Q}^t, \mathbf{K}^t, \mathbf{V}^t), \\ &= \mathrm{Attention}(\mathbf{X}_S^i \mathbf{W}_Q^t, \mathbf{K}^t, \mathbf{V}^t), \end{aligned}$$

Attention

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

Adaptive Spectral Prompts (ASP)

We propose ASP to adapt spectral prompts with spectral information



Attention

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

Query

$$\mathbf{W}_Q^t \in \mathbb{R}^{d_{i-1}\times d_k} \quad \mathbf{Q}^t = \mathbf{X}_S^i \mathbf{W}_Q^t \in \mathbb{R}^{n\times d_k}$$

To reduce complexity we downsample Q and K

Key

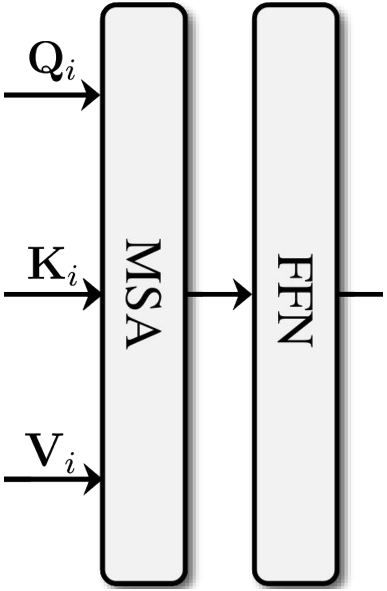
$$\hat{\mathbf{K}}^t = \text{Reshape}\left(\frac{n}{r}, d_{i-1}\cdot r\right)(\mathbf{X}_S^i),$$
$$\mathbf{K}^t = \hat{\mathbf{K}}^t\mathbf{W}_K^t \in \mathbb{R}^{\frac{n}{r}\times d_k}$$

Value

$$\hat{\mathbf{V}}^t = \text{Reshape}\left(\frac{n}{r}, d_{i-1}\cdot r\right)(\mathbf{X}_S^i),$$
$$\mathbf{V}^t = \hat{\mathbf{V}}^t\mathbf{W}_V^t \in \mathbb{R}^{\frac{n}{r}\times d_v}$$

Adaptive Spectral Prompts (ASP)

We propose ASP to adapt spectral prompts with spectral information

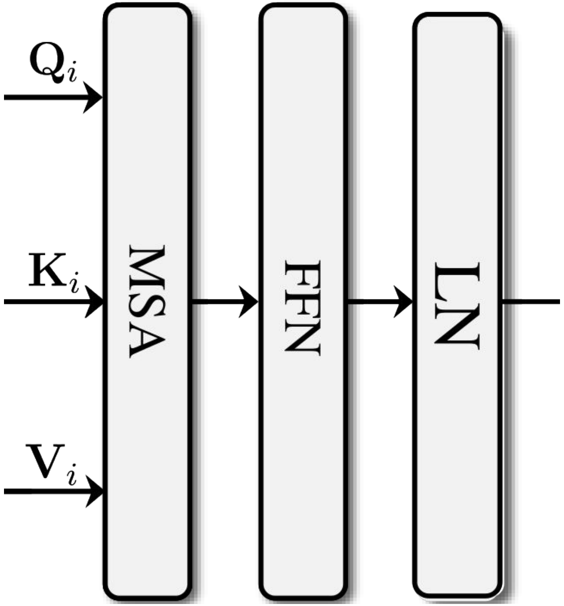


Feed-Forward Network

$$\mathcal{Z}^i = \text{FFN}^i_\psi(\mathcal{A}^i) = \text{MLP}^i(\text{GELU}^i(\text{MLP}^i(\mathcal{A}^i)))+\mathcal{A}^i$$

Adaptive Spectral Prompts (ASP)

We propose ASP to adapt spectral prompts with spectral information

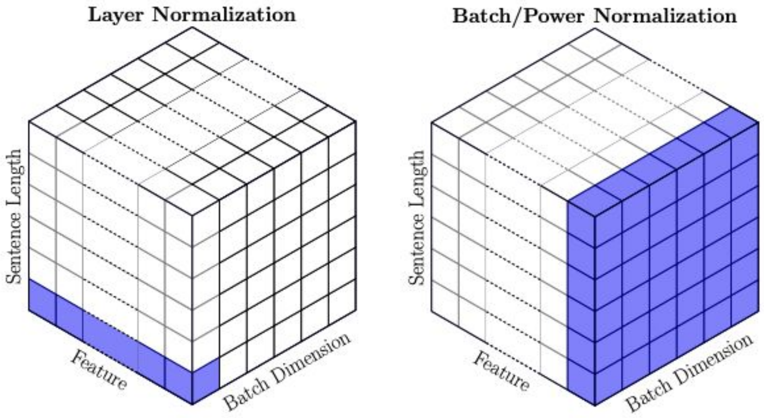


Layer Normalization

$$[\hat{\mathbf{P}}^i, \mathbf{E}_s^i] = \mathrm{LN}^i(\mathcal{Z}_k^i)$$

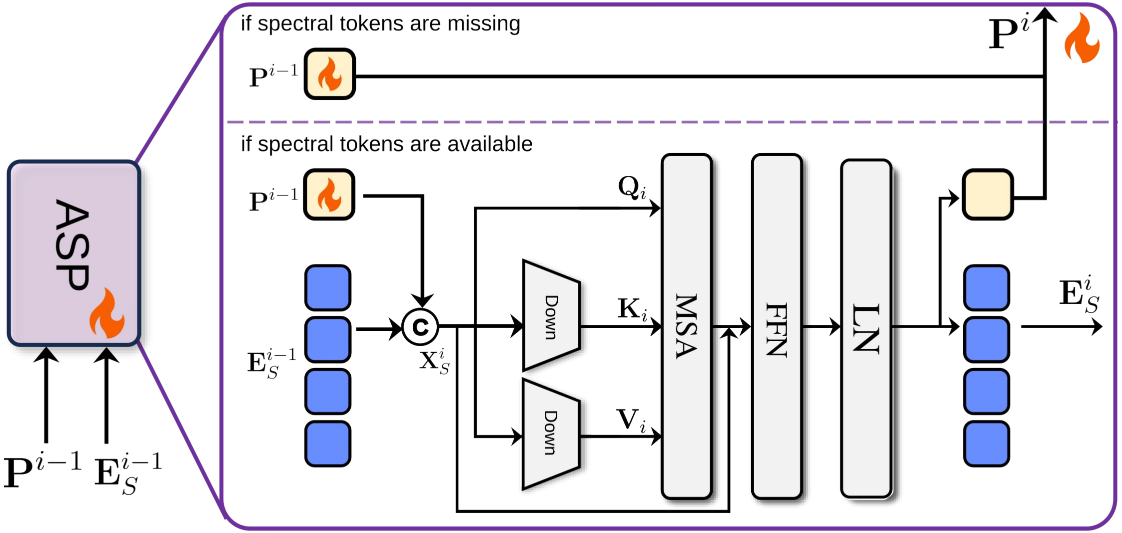
Normalizes across

feature dimension



Adaptive Spectral Prompts (ASP)

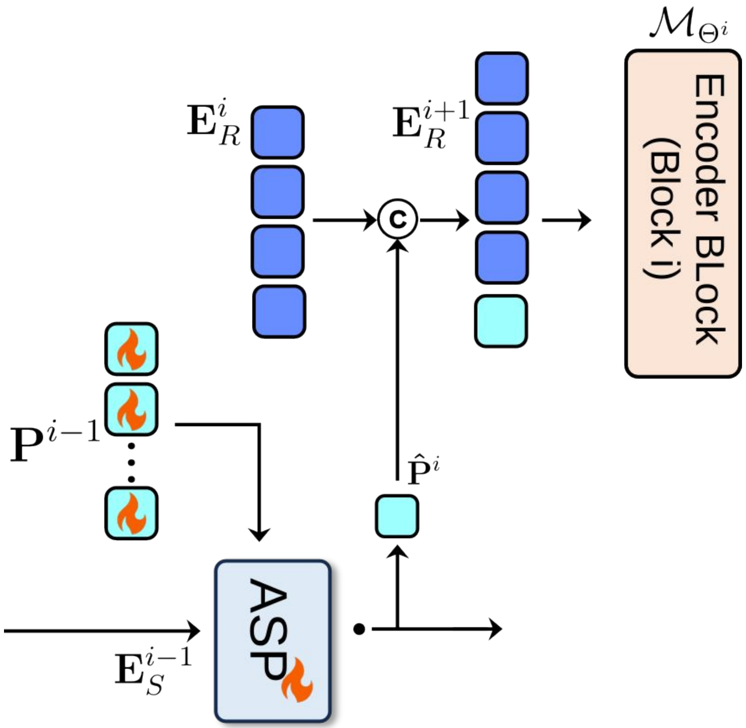
We propose ASP to adapt spectral prompts with spectral information



$$[\hat{\mathbf{P}}^i, \mathbf{E}_s^i] = \mathrm{LN}(\mathrm{FFN}_\delta^i(\mathrm{MSA}_\psi^i(\mathbf{X}_S^i)))$$

Adaptive Spectral Prompts (ASP)

The output of the ASP module will be concatenated with RGB tokens



$$[\hat{\mathbf{P}}^i, \mathbf{E}_S^i] = \mathrm{ASP}_i([\mathbf{P}^{i-1}; \mathbf{E}_S^{i-1}])$$

Prompts + RGB tokens

$$\mathbf{X}_R^i = [\mathbf{P}^i; \mathbf{E}_R^{i-1}] \in \mathbb{R}^{(l+n)\times d}$$

Output of model block i

$$[\_, \mathbf{E}_R^{i+1}] = \mathcal{M}_{\Theta^i}(\mathbf{X}_R^i)$$

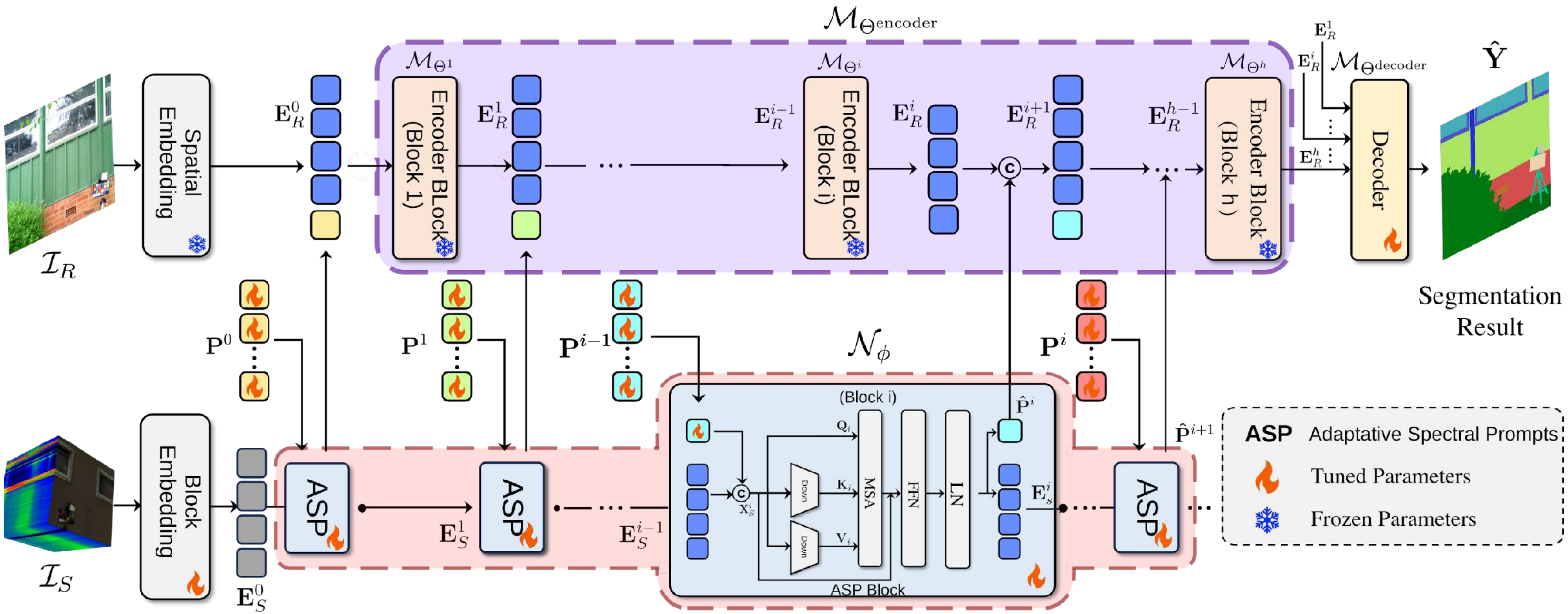
Input for next ASP block

$$\mathbf{X}_R^i = [\hat{\mathbf{P}}^i; \mathbf{E}_R^{i-1}] \in \mathbb{R}^{(l+n)\times d}$$

Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H \times W \times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H \times W \times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H \times W}$

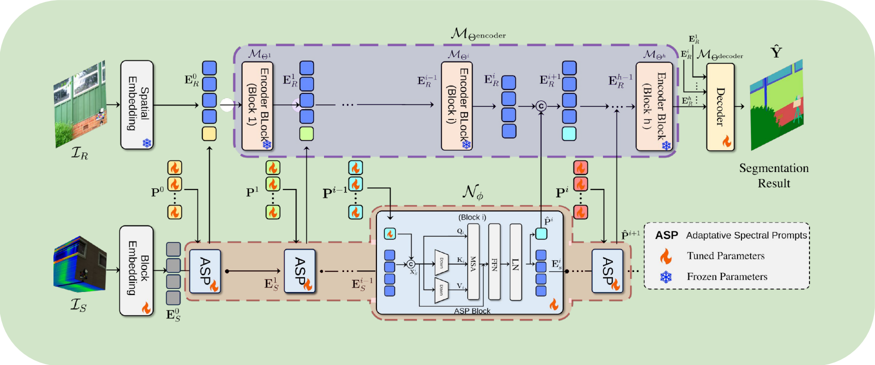
$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$

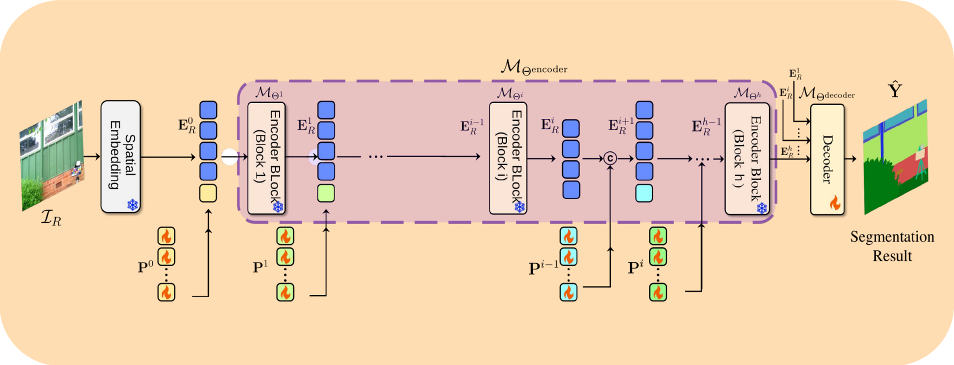


Modality Dropout

Given a dataset: $\mathcal{D} = (\hat{\mathcal{I}}_S^i, \mathcal{I}_R^i, \mathbf{Y}^i)_{i=1}^N$, modality dropout allows the model to be able to handle missing spectral situations during inference

$$\mathcal{I}_S = \begin{cases} \hat{\mathcal{I}}_S, & \text{con probabilidad} \quad 1 - p_d \\ \emptyset, & \text{con probabilidad} \quad p_d \end{cases}$$

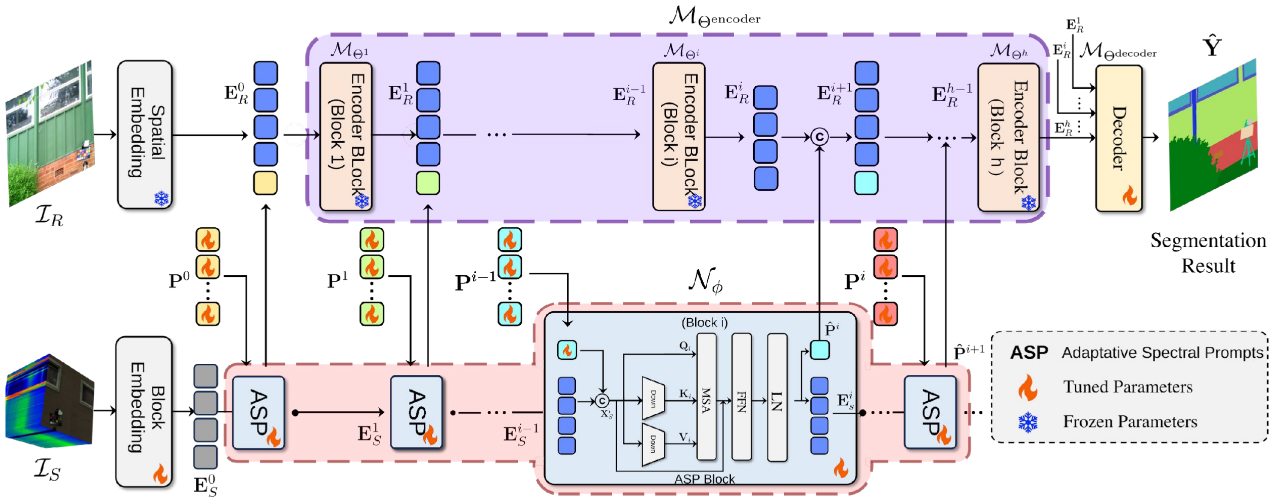




Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H\times W\times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H\times W\times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H\times W}$

$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$

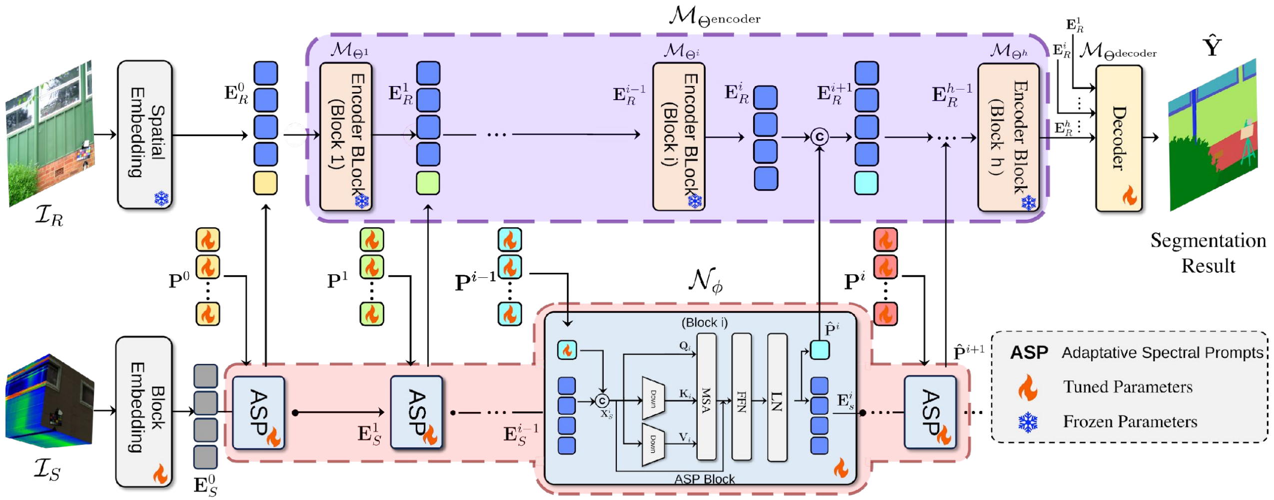


2.Diseñar una arquitectura de transformer de visión que integre información espectral y de color (RGB) de una escena para segmentarla en distintos materiales

Architecture overview

Spectral image: $\mathcal{I}_S \in \mathbb{R}^{H \times W \times B}$ RGB image: $\mathcal{I}_R \in \mathbb{R}^{H \times W \times 3}$ Prediction: $\hat{\mathbf{Y}} \in \{1, ..., c\}^{H \times W}$

$$\hat{\mathbf{Y}} = f\left(\mathcal{M}_\theta(\mathcal{I}_R), \mathcal{N}_\phi(\mathcal{I}_S, \mathcal{P})\right)$$



2. Diseñar una arquitectura de transformer de visión que integre información espectral y de color (RGB) de una escena para segmentarla en distintos materiales

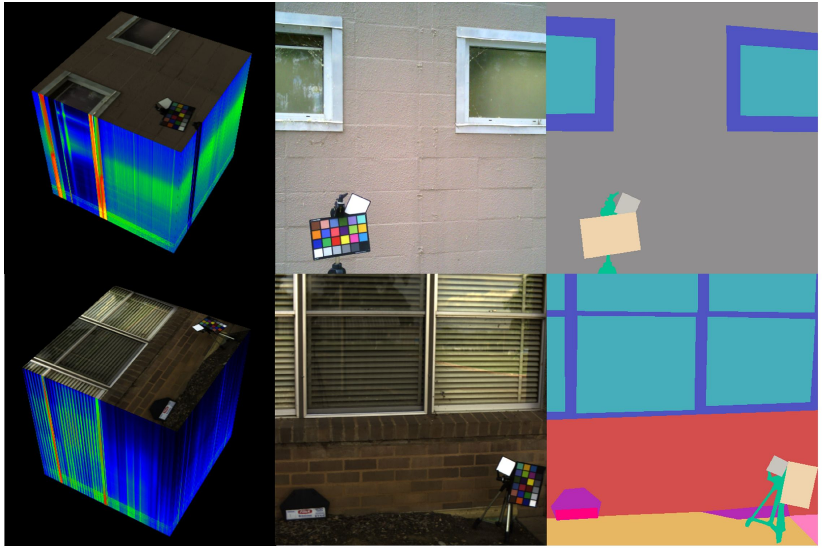
Simulations and Results

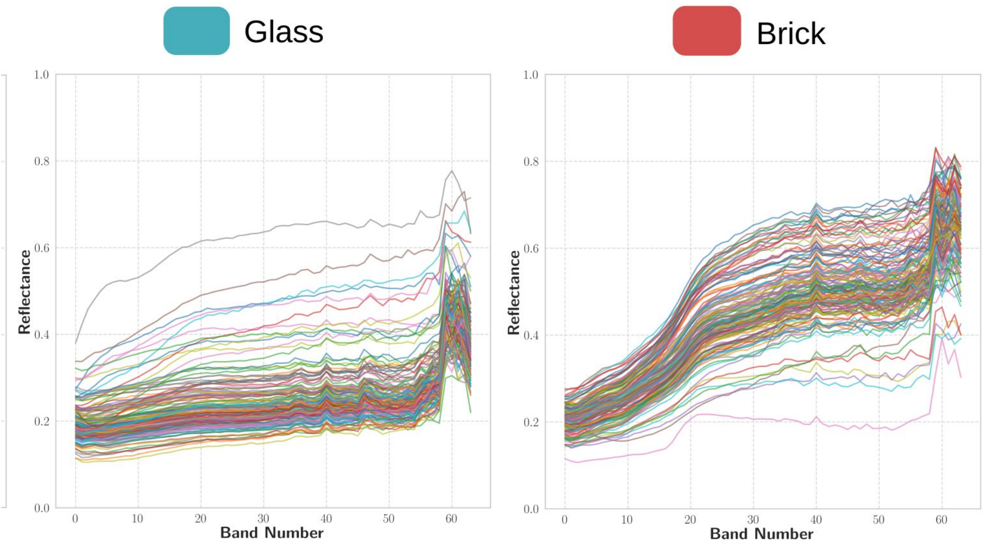
Universidad Industrial de Santander

Dataset

LIB-HSI is used as the default dataset for our experiments

$$N = 513$$





has a spatial resolution of 512x512 pixels and 204 bands, spectral range from 400 to 1000 nm, 44 classes

To reduce computational complexity, we downsample to 64 bands using a moving average

LIB-HSI-Fixed

We found that some classes were misrepresented in some splits in the dataset

Class	Images	Number of pixels		
		Train	Validation	Test
Wood Ground	1	116257	0	0
Door-plastic	2	177131	0	0

Then, we remove this classes and images, and we propose a new split for the dataset called: LIB-HSI-FIXED

LIB-HSI-Fixed

We found that some classes were misrepresented in some splits in the dataset

1. Identificar y seleccionar bases de datos de imágenes espectrales y de color (RGB) adecuadas para el entrenamiento y prueba del algoritmo,

Class	Images	Number of pixels		
		Train	Validation	Test
Wood Ground	1	116257	0	0
Door-plastic	2	177131	0	0

Then, we remove this classes and images, and we propose a new split for the dataset called: LIB-HSI-FIXED

LIB-HSI-Fixed

We found that some classes were misrepresented in some splits in the dataset

1. Identificar y seleccionar bases de datos de imágenes espectrales y de color (RGB) adecuadas para el entrenamiento y prueba del algoritmo,

Class	Images	Number of pixels		
		Train	Validation	Test
Wood Ground	1	116257	0	0
Door-plastic	2	177131	0	0

Then, we remove this classes and images, and we propose a new split for the dataset called: LIB-HSI-FIXED

Metrics

We use the same metrics as SOTA, this is: absolute accuracy and MIoU

Absolute Accuracy

$$\mathrm{Acc} = \frac{\sum_{c=1}^{C} (TP)_c}{M}$$

Mean Intersection over Union

$$\mathrm{IoU}_c = \frac{(TP)_c}{(TP)_c + (FP)_c + (FN)_c},$$

$$\text{mean IoU} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{IoU}_c$$

Simulations Setup

We validate the benefits of our proposed framework. We used segformer-3 as our base transform encoder-decoder

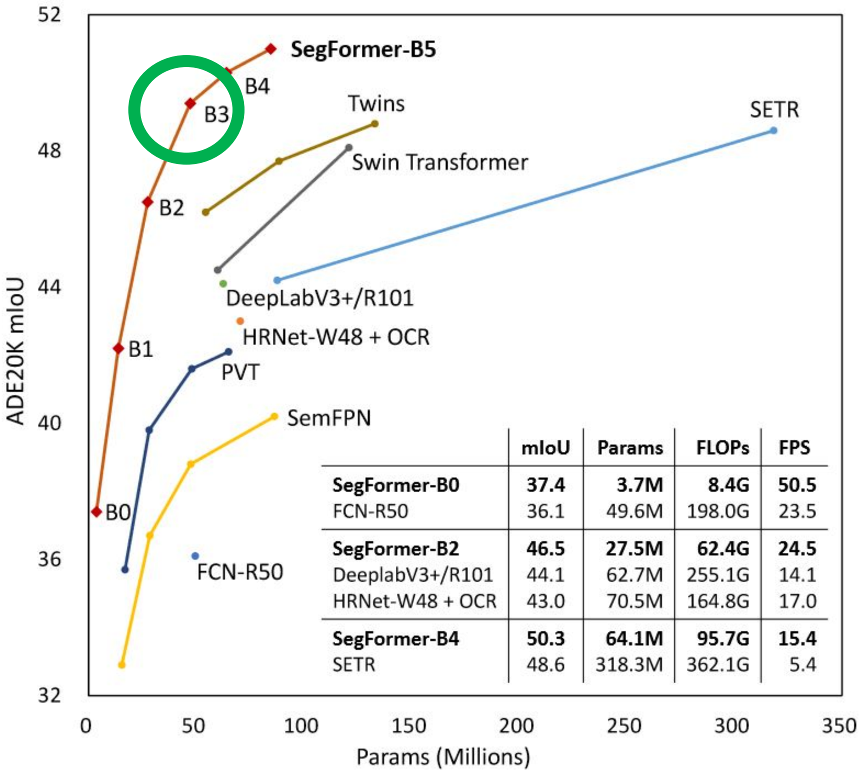
$$p_h = 8 \quad p_w = 8 \quad p_b = 16$$

$$p_{h'} = 4 \quad p_{w'} = 4$$

$$r = [64, 16, 4, 1] \qquad p_d = 0.2$$

Lr-warm-up and cosine-scheduler for all experiments

We set Data Augmentations like: rotation, flip and shifting



	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G	5.4

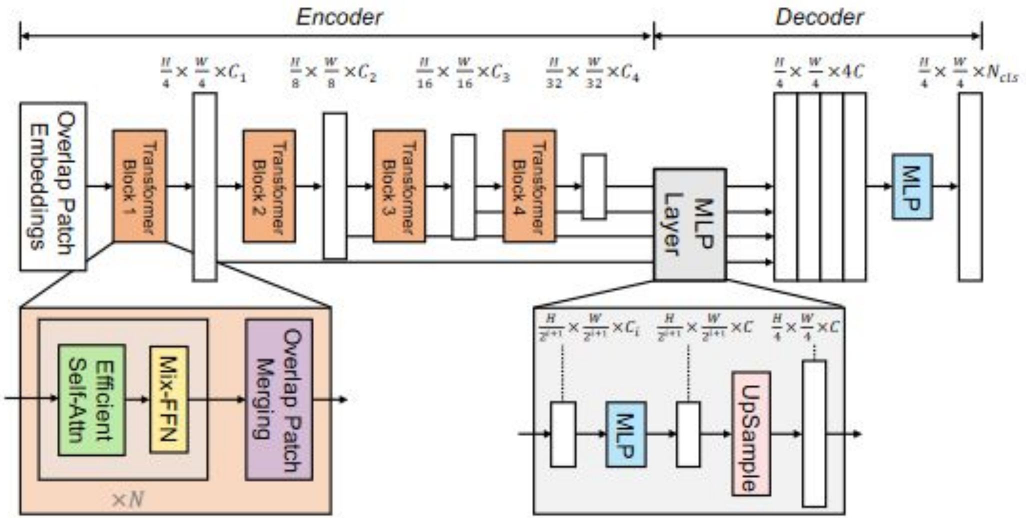
Simulations Setup

We validate the benefits of our proposed framework. We used segformer-3 as our base transform encoder-decoder

$$p_h = 8 \quad p_w = 8 \quad p_b = 16$$

$$p_{h'} = 4 \quad p_{w'} = 4$$

$$r = [64, 16, 4, 1] \qquad p_d = 0.2$$



Lr-warm-up and cosine-scheduler for all experiments

We set Data Augmentations like: rotation, flip and shifting

We trained all models on a GPU RTX 3090 over 200 epochs

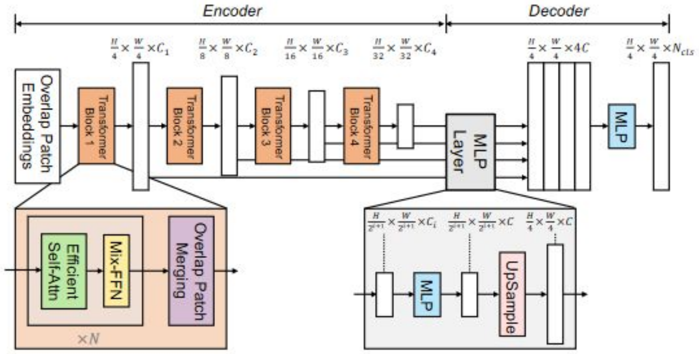
Simulations Setup

We validate the benefits of our proposed framework. We used segformer-3 as our base transform encoder-decoder

$$p_h = 8 \quad p_w = 8 \quad p_b = 16$$

$$p_{h'} = 4 \quad p_{w'} = 4$$

$$r = [64, 16, 4, 1] \qquad p_d = 0.2$$



Lr-warm-up and cosine-scheduler for all experiments

We set Data Augmentations like: rotation, flip and shifting

3. Implementar en Python la arquitectura de transformer de visión diseñada para la segmentación de los materiales de una escena.

We trained all models on a GPU RTX 3090 over 200 epochs

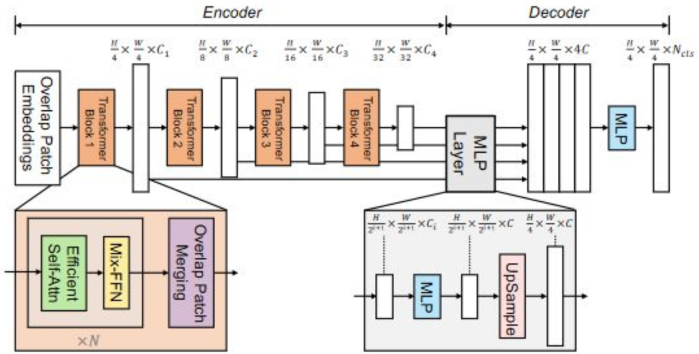
Simulations Setup

We validate the benefits of our proposed framework. We used segformer-3 as our base transform encoder-decoder

$$p_h = 8 \quad p_w = 8 \quad p_b = 16$$

$$p_{h'} = 4 \quad p_{w'} = 4$$

$$r = [64, 16, 4, 1] \qquad p_d = 0.2$$



Lr-warm-up and cosine-scheduler for all experiments

We set Data Augmentations like: rotation, flip and shifting

3. Implementar en Python la arquitectura de transformer de visión diseñada para la segmentación de los materiales de una escena.

We trained all models on a GPU RTX 3090 over 200 epochs

Ablation Studies

We validate the benefits of our proposed framework

Method	Acc	Average per class IoU	Parameters
Full Fine-tuning	85,94	46,98	44,7 millones
Prompt-tuning RGB	86,4	54,92	3,3 millones
Ours *(w/o modality dropout)*	88,16	54,95	11 millones
Ours *(Prompt Tuning Spectral)*	**88,54**	**56,84**	11 millones

Ablation with Missing Modality

We validate both cases, when spectral is available and when is not.

Input	Acc	Average per clasas IoU
RGB	88,54	56,84
RGB + Spectral	**88,63**	**56,95**

Ablation with Missing Modality

We validate both cases, when spectral is available and when is not.

Input	Acc	Average per clasas
		IoU
RGB	88,54	56,84
RGB + Spectral	**88,63**	**56,95**

4. Evaluar el desempeño del algoritmo desarrollado mediante métricas de rendimiento estándar en el área de segmentación.

Ablation with Missing Modality

We validate both cases, when spectral is available and when is not.

Input	Acc	Average per clasas IoU
RGB	88,54	56,84
RGB + Spectral	**88,63**	**56,95**

4. Evaluar el desempeño del algoritmo desarrollado mediante métricas de rendimiento estándar en el área de segmentación.

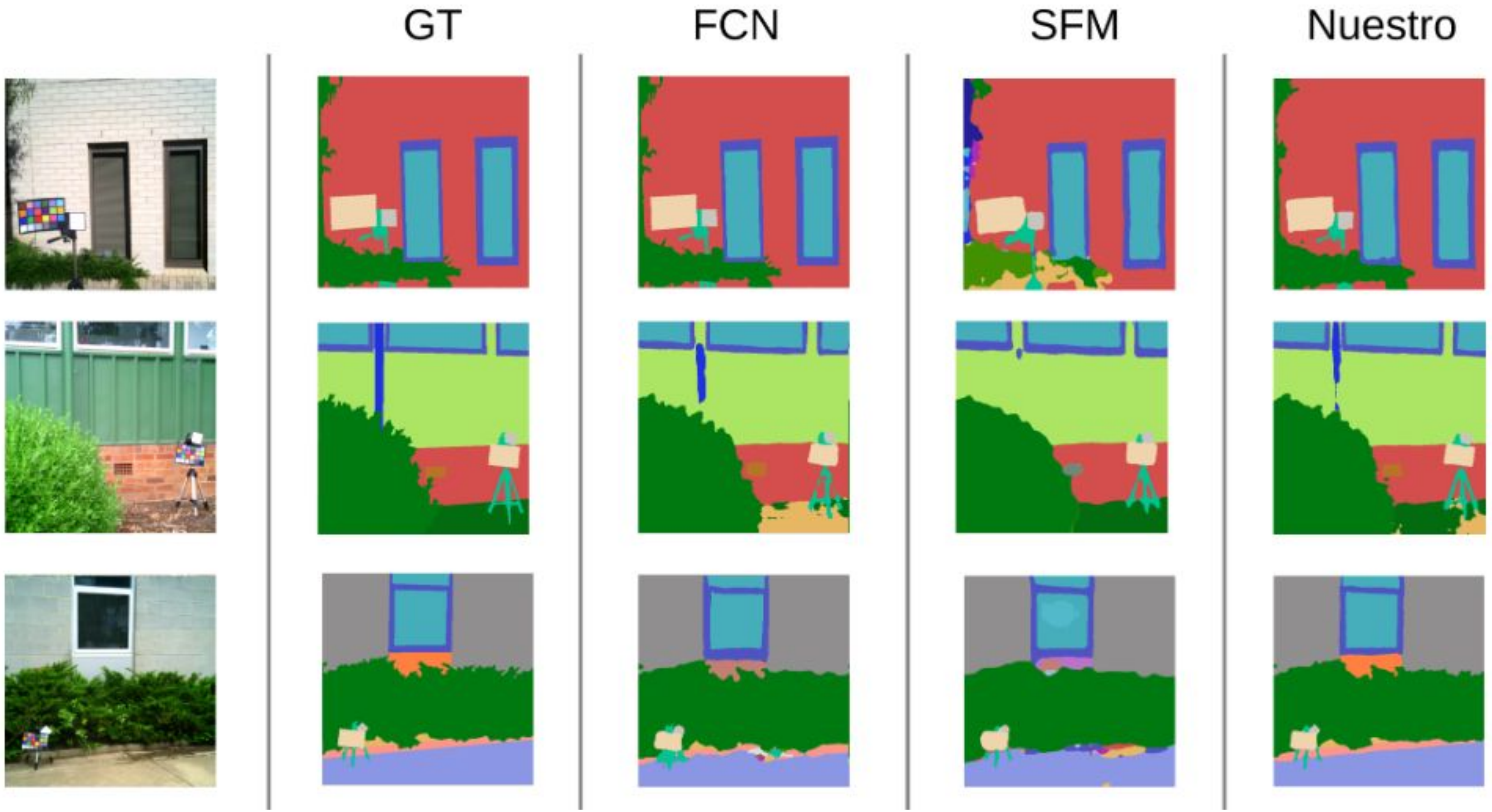
Comparison against SOTA

Comparison against SOTA

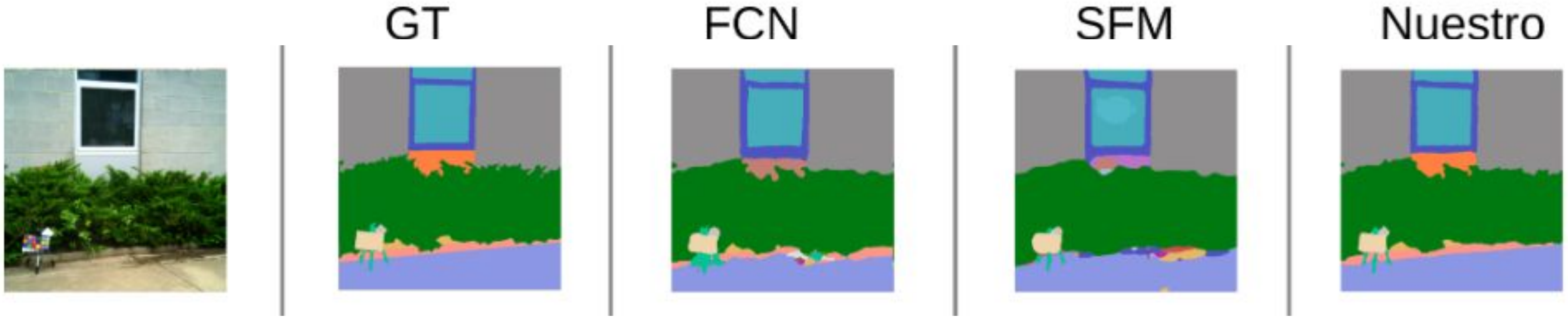
We compare with state-of-the-art methods

Method	Acc	Average per class IoU
FCN[112]	$82,9$	$44,3$
SFM+HRnet[113]	$86,47$	$48,37$
CSSF+DeepLabV3[114]	$---$	$51,2$
Ours (Prompt Tuning Spectral)	$\mathbf{88{,}36}$	$\mathbf{53,28}$

Visual Results



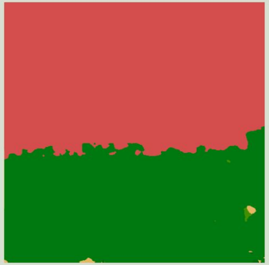
Visual Results

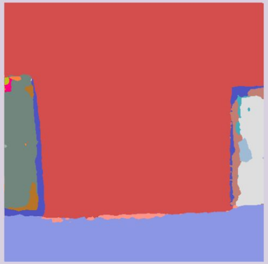


Experimental Results

RGB

Segmentation

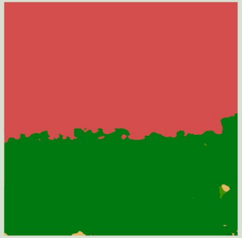
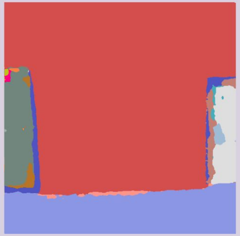




- Brick wall
- Vegetation plant
- Soil
- Vegetation ground
- Concrete ground
- Concrete wall
- Metal Profiled door

Experimental Results

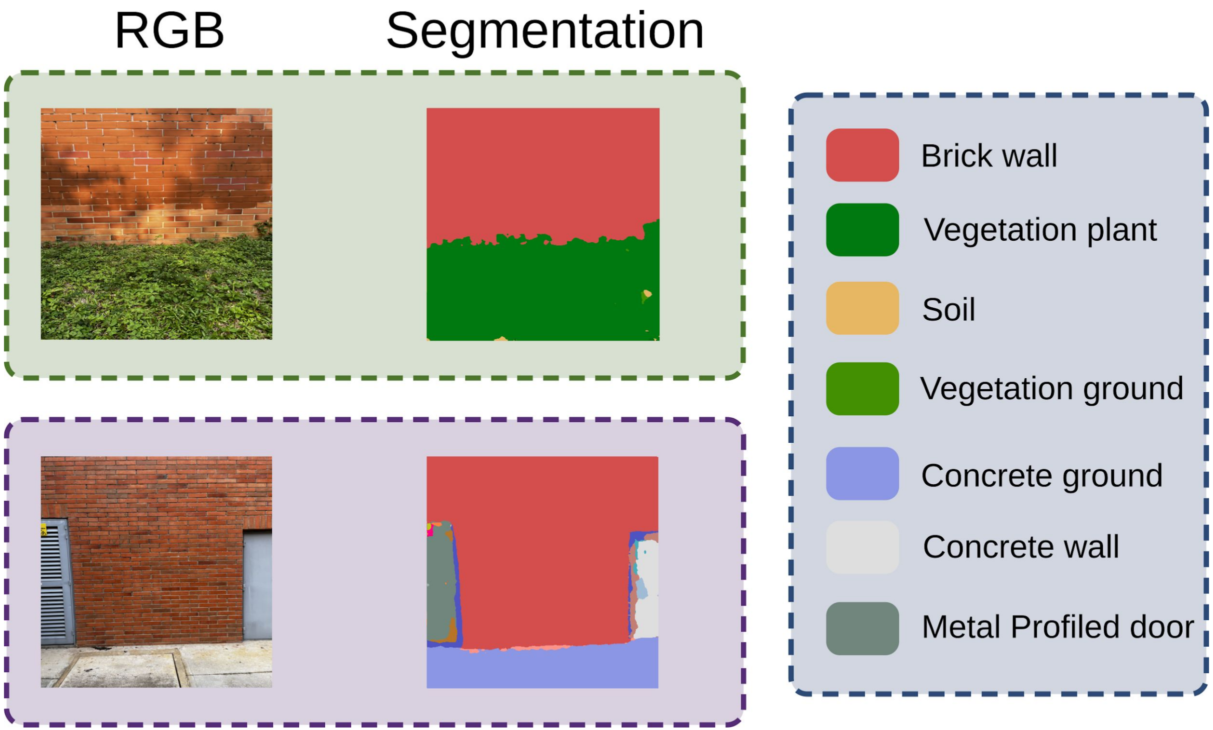
5. Validar cualitativamente el algoritmo sobre un conjunto de imágenes de color adquiridas con una cámara disponible en dispositivos electrónicos de consumo.

RGB	Segmentation
	
	

- Brick wall
- Vegetation plant
- Soil
- Vegetation ground
- Concrete ground
- Concrete wall
- Metal Profiled door

Experimental Results

5. Validar cualitativamente el algoritmo sobre un conjunto de imágenes de color adquiridas con una cámara disponible en dispositivos electrónicos de consumo.



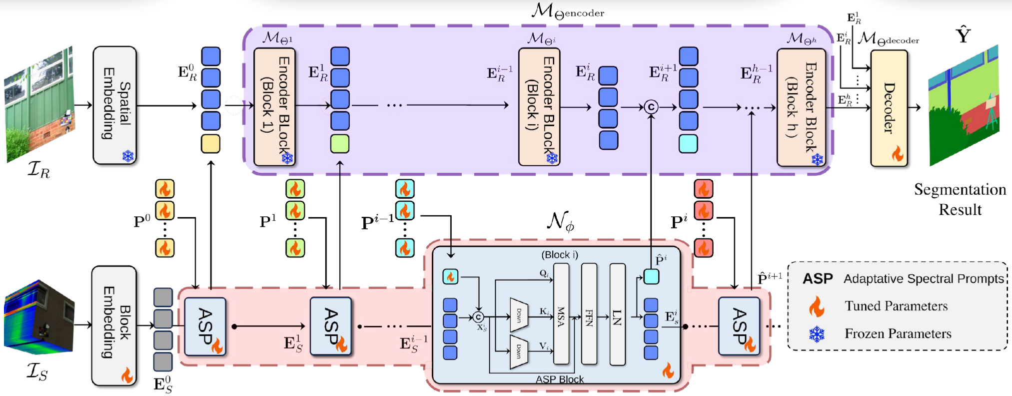
Conclusions

Conclusions

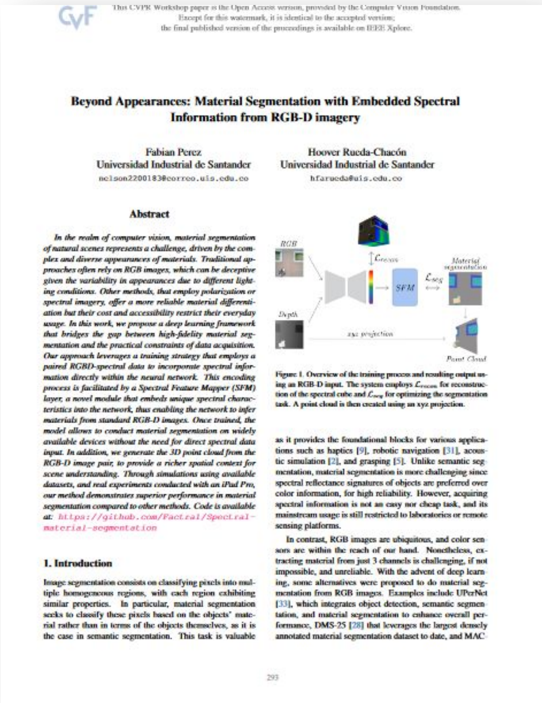
Prompt tuning proves to be a highly efficient technique, enabling effective model adaptation with minimal trainable parameters.

The integration of spectral information significantly enhances the quality of material segmentation, providing more accurate and detailed results.

Embedded spectral information in the network architecture allows for robust performance even in scenarios where spectral data is unavailable during inference.



Additional Impact



Beyond Appearances: Material Segmentation with Embedded Spectral Information from RGB-D imagery

Fabian Perez
Universidad Industrial de Santander
nelson2200183@correo.uis.edu.co

Hoover Rueda-Chacón
Universidad Industrial de Santander
hfarueda@uis.edu.co

Abstract

In the realm of computer vision, material segmentation of natural scenes represents a challenge, driven by the complex and diverse appearances of materials. Traditional approaches often rely on RGB images, which can be deceptive given the variability in appearances due to different lighting conditions. Other methods, that employ polarization or spectral imagery, offer a more reliable material differentiation but their cost and accessibility restrict their everyday usage. In this work, we propose a deep learning framework that bridges the gap between high-fidelity material segmentation and the practical constraints of data acquisition. Our approach leverages a training strategy that employs a paired RGBD-spectral data to incorporate spectral information directly within the neural network. This encoding process is facilitated by a Spectral Feature Mapper (SFM) layer, a novel module that embeds unique spectral characteristics into the network, thus enabling the network to infer materials from standard RGB-D images. Once trained, the model allows to conduct material segmentation on widely available devices without the need for direct spectral data input. In addition, we generate the 3D point cloud from the RGB-D image pair, to provide a richer spatial context for scene understanding. Through simulations using available datasets, and real experiments conducted with an iPad Pro, our method demonstrates superior performance in material segmentation compared to other methods. Code is available at: https://github.com/Factral/Spectral-material-segmentation

1. Introduction

Image segmentation consists on classifying pixels into multiple homogeneous regions, with each region exhibiting similar properties. In particular, material segmentation seeks to classify these pixels based on the objects' material rather than in terms of the objects themselves, as it is the case in semantic segmentation. This task is valuable

Figure 1. Overview of the training process and resulting output using an RGB-D input. The system employs $\mathcal{L}_{recon}$ for reconstruction of the spectral cube and $\mathcal{L}_{seg}$ for optimizing the segmentation task. A point cloud is then created using an xyz projection.

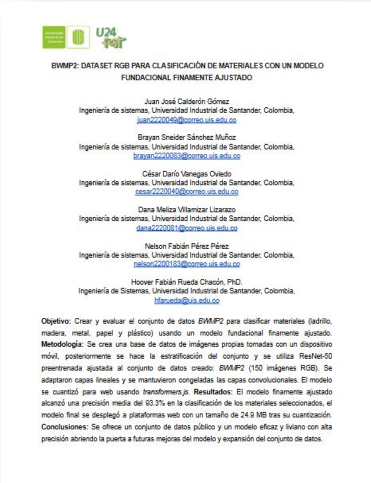
as it provides the foundational blocks for various applications such as haptics [9], robotic navigation [31], acoustic simulation [2], and grasping [5]. Unlike semantic segmentation, material segmentation is more challenging since spectral reflectance signatures of objects are preferred over color information, for high reliability. However, acquiring spectral information is not an easy nor cheap task, and its mainstream usage is still restricted to laboratories or remote sensing platforms.

In contrast, RGB images are ubiquitous, and color sensors are within the reach of our hand. Nonetheless, extracting material from just 3 channels is challenging, if not impossible, and unreliable. With the advent of deep learning, some alternatives were proposed to do material segmentation from RGB images. Examples include UPerNet [33], which integrates object detection, semantic segmentation, and material segmentation to enhance overall performance, DMS-25 [28] that leverages the largest densely annotated material segmentation dataset to date, and MAC-

293

HoCV

Semillero Hands-On Computer Vision



BWMP2: DATASET RGB PARA CLASIFICACIÓN DE MATERIALES CON UN MODELO FUNDACIONAL FINAMENTE AJUSTADO

Juan José Calderón Gómez
Ingeniería de sistemas, Universidad Industrial de Santander, Colombia,
juan2220049@correo.uis.edu.co

Brayan Sneider Sánchez Muñoz
Ingeniería de sistemas, Universidad Industrial de Santander, Colombia,
brayan2220083@correo.uis.edu.co

César Darío Vanegas Oviedo
Ingeniería de sistemas, Universidad Industrial de Santander, Colombia,
cesar2220040@correo.uis.edu.co

Dana Meliza Villamizar Lizarazo
Ingeniería de sistemas, Universidad Industrial de Santander, Colombia,
dana2220081@correo.uis.edu.co

Nelson Fabián Pérez Pérez
Ingeniería de sistemas, Universidad Industrial de Santander, Colombia,
nelson2200183@correo.uis.edu.co

Hoover Fabián Rueda Chacón, PhD.
Ingeniería de Sistemas, Universidad Industrial de Santander, Colombia,
hfarueda@uis.edu.co

Objetivo: Crear y evaluar el conjunto de datos *BWMP2* para clasificar materiales (ladrillo, madera, metal, papel y plástico) usando un modelo fundacional finamente ajustado. **Metodología:** Se crea una base de datos de imágenes propias tomadas con un dispositivo móvil, posteriormente se hace la estratificación del conjunto y se utiliza ResNet-50 preentrenada ajustada al conjunto de datos creado: *BWMP2* (150 imágenes RGB). Se adaptaron capas lineales y se mantuvieron congeladas las capas convolucionales. El modelo se cuantizó para web usando *transformers.js*. **Resultados:** El modelo finamente ajustado alcanzó una precisión media del 93.3% en la clasificación de los materiales seleccionados, el modelo final se desplegó a plataformas web con un tamaño de 24.9 MB tras su cuantización. **Conclusiones:** Se ofrece un conjunto de datos público y un modelo eficaz y liviano con alta precisión abriendo la puerta a futuras mejoras del modelo y expansión del conjunto de datos.

U24FEST Paper

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Remote Internship at KAUST

To be submitted to CVPR 2025

Thanks