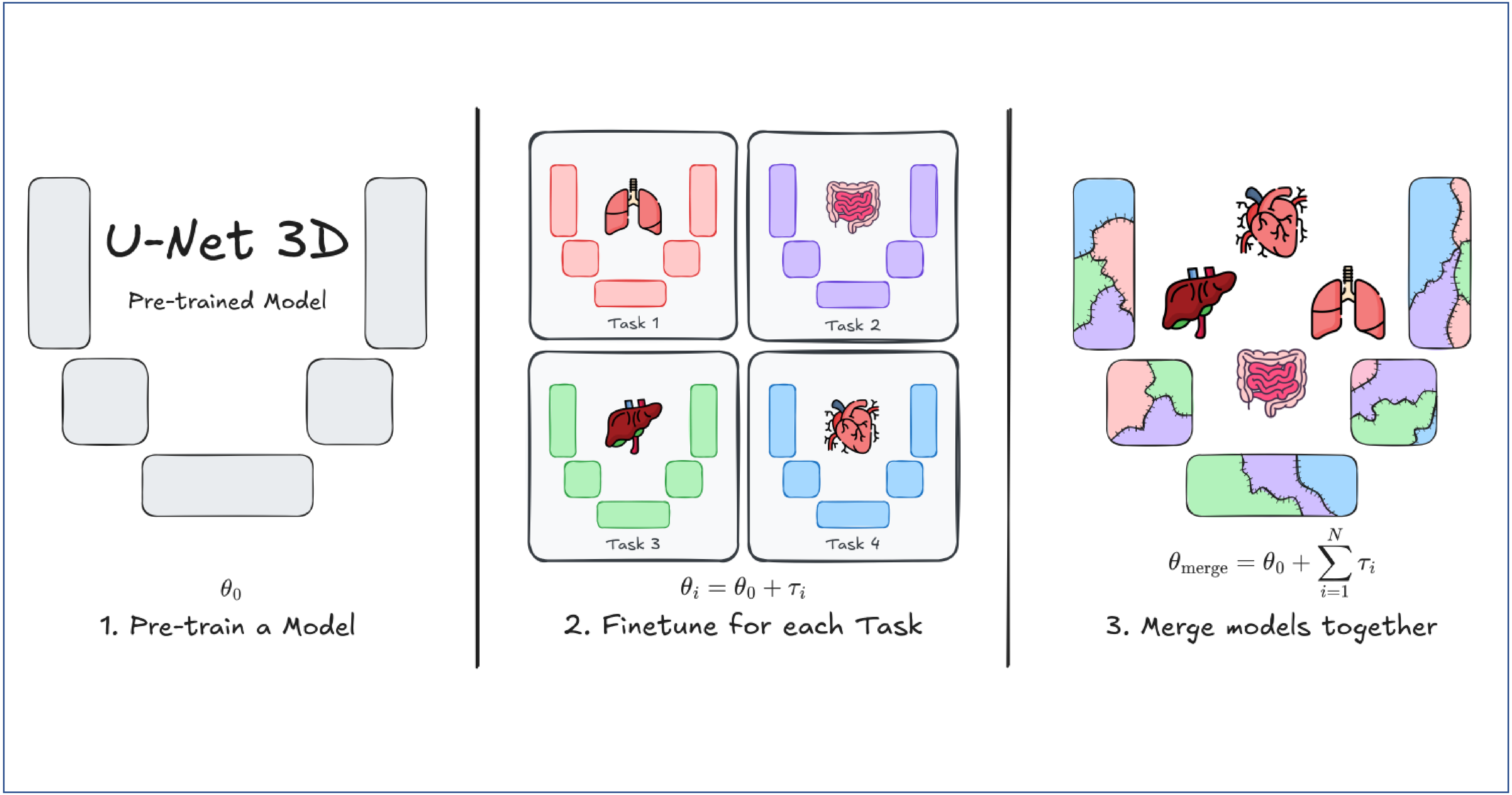
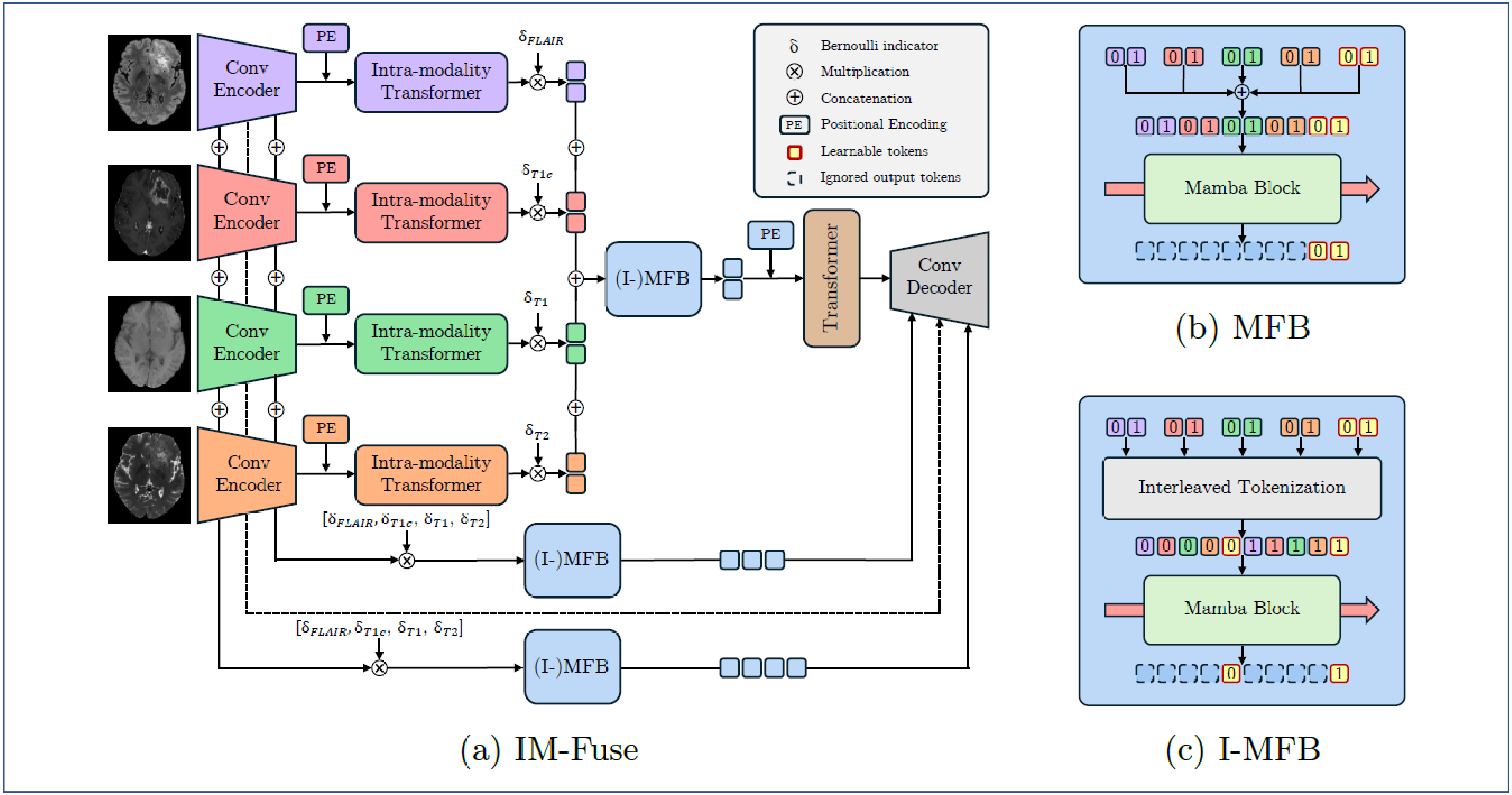

U-Net 3D
Pre-trained Model
θ_0
1. Pre-train a Model
Task 1
Task 2
Task 3
Task 4
$\theta_i = \theta_0 + \tau_i$
2. Finetune for each Task
$\theta_{\mathrm{merge}} = \theta_0 + \sum_{i=1}^{N} \tau_i$
3. Merge models together



PE
Conv Encoder
Intra-modality Transformer
δFLAIR
δT1c
δT1
δT2
(I-)MFB
Transformer
Conv Decoder
[δFLAIR, δT1c, δT1, δT2]
δ Bernoulli indicator
⊗ Multiplication
⊕ Concatenation
PE Positional Encoding
Learnable tokens
Ignored output tokens
Mamba Block
Interleaved Tokenization
(a) IM-Fuse
(b) MFB
(c) I-MFB

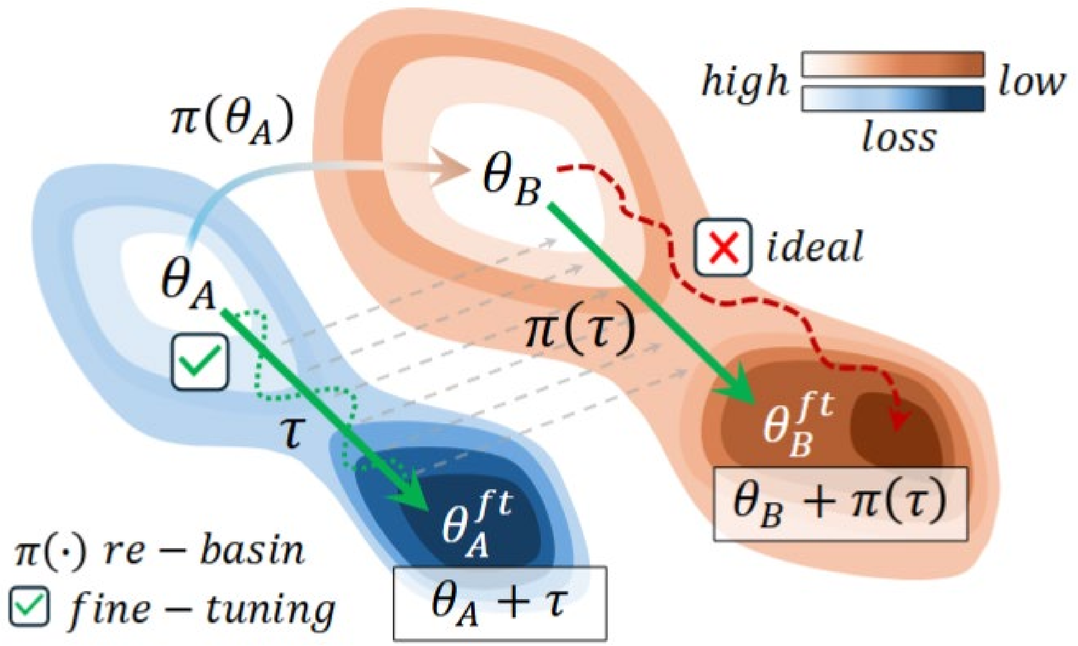
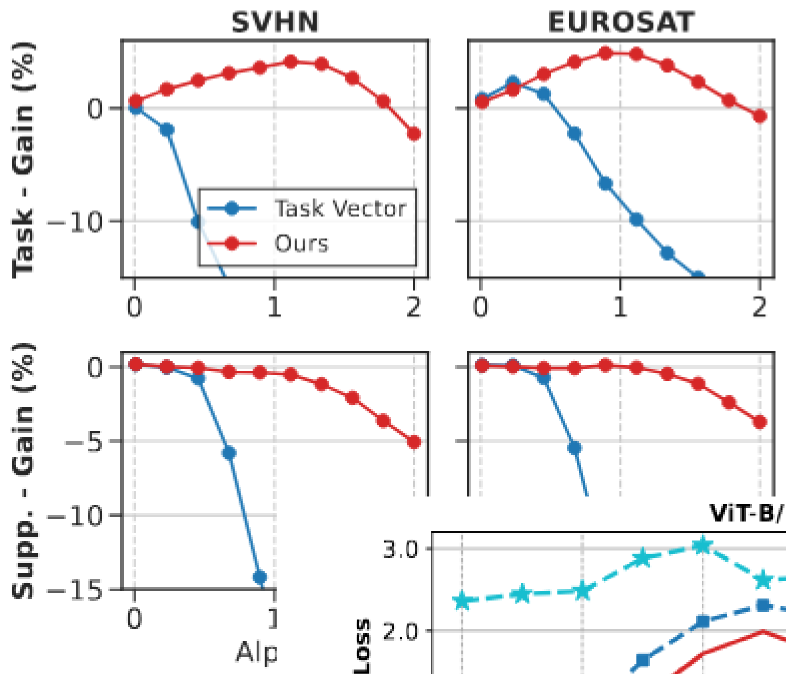


Figure 1. Transporting task vector τ from a fine-tuned base model $\theta_A^{ft} = \theta_A + \tau$ to a new release θ_B.

Method	EUROSAT		DTD		GTSRB		SVHN	
	Task	Supp.	Task	Supp.	Task	Supp.	Task	Supp.
θ_B *zero-shot*	49.02	68.73	47.50	68.73	43.42	68.73	45.97	68.73
$\theta_B + \tau$	-7.62	-16.15	-0.15	-0.10	-5.39	-0.70	-22.00	-16.45
$\theta_B + \pi(\tau)$ *(Optimal Transport)*	-14.05	-5.28	-0.53	-1.18	-2.43	-1.30	-12.30	-2.70
$\theta_B + \pi(\tau)$ *(GiT Re-Basin)*	+0.95	-0.48	-0.91	**-0.02**	+0.76	**-0.05**	+0.79	**+0.30**
TransFusion (Ours)	**+4.95**	**-0.06**	**+0.21**	-0.08	**+1.10**	-0.40	**+3.64**	-0.48

Figure 4. Zero-shot gain/drop relative to θ_B of task transport $\theta_B + \alpha\tau$ (blue) and our strategy $\theta_B + \alpha\pi(\tau)$ (red) varying α.



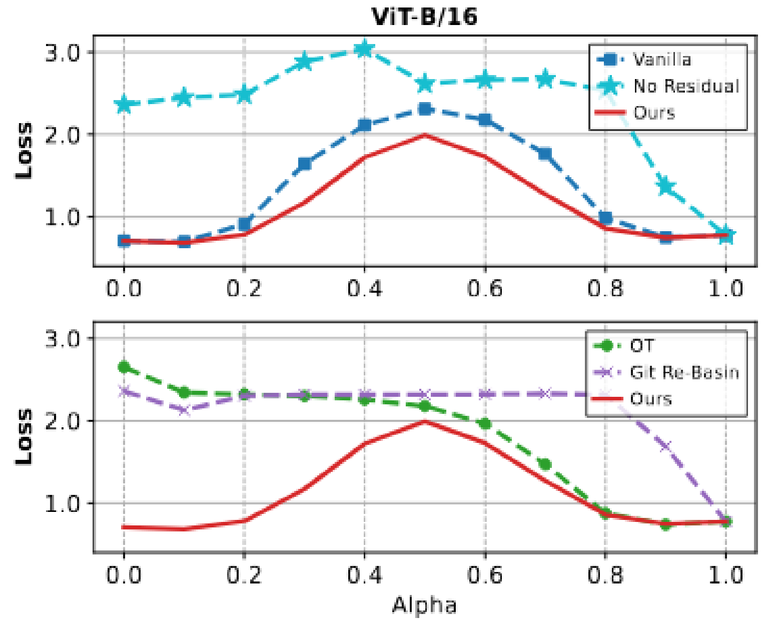
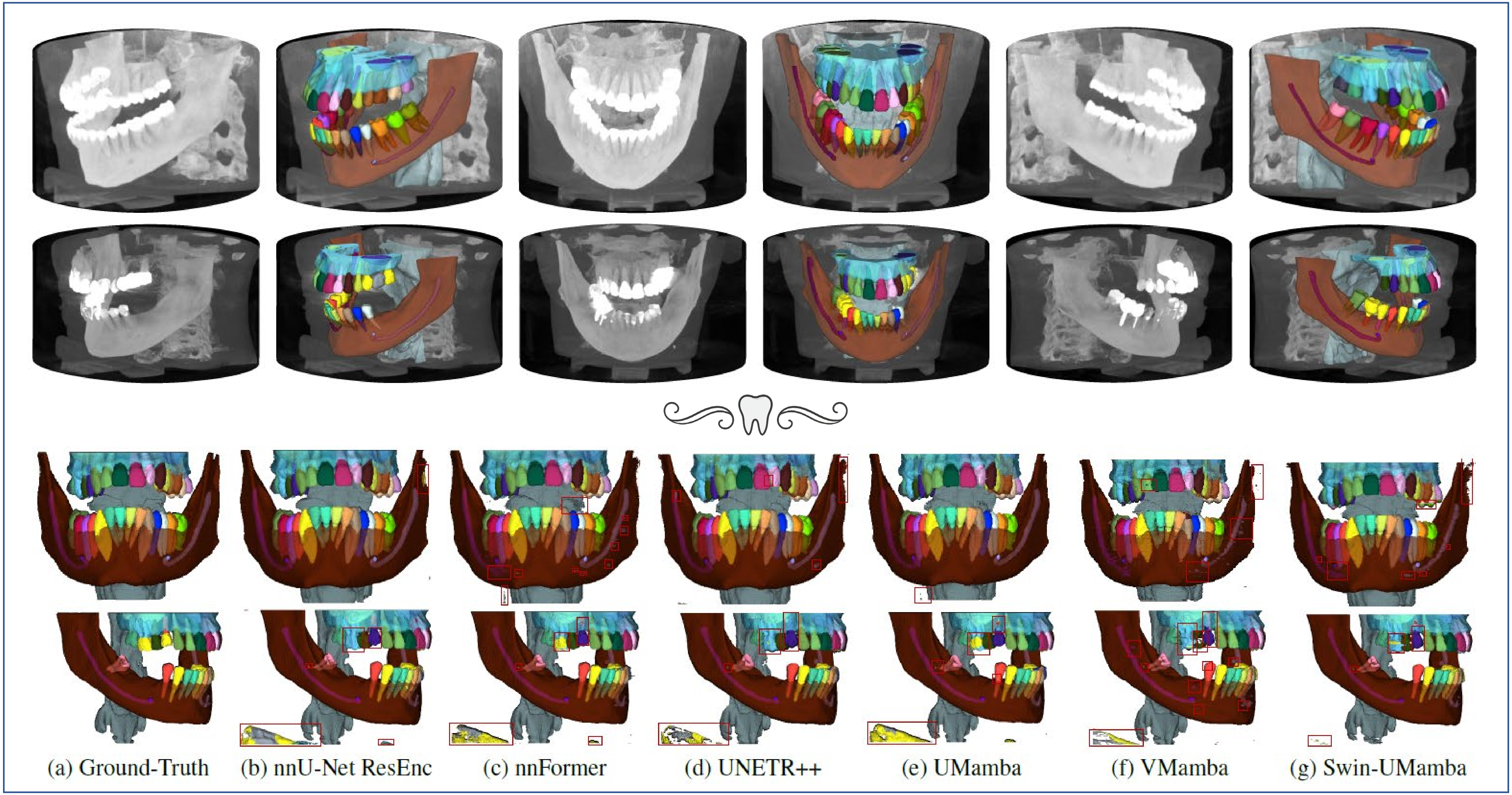


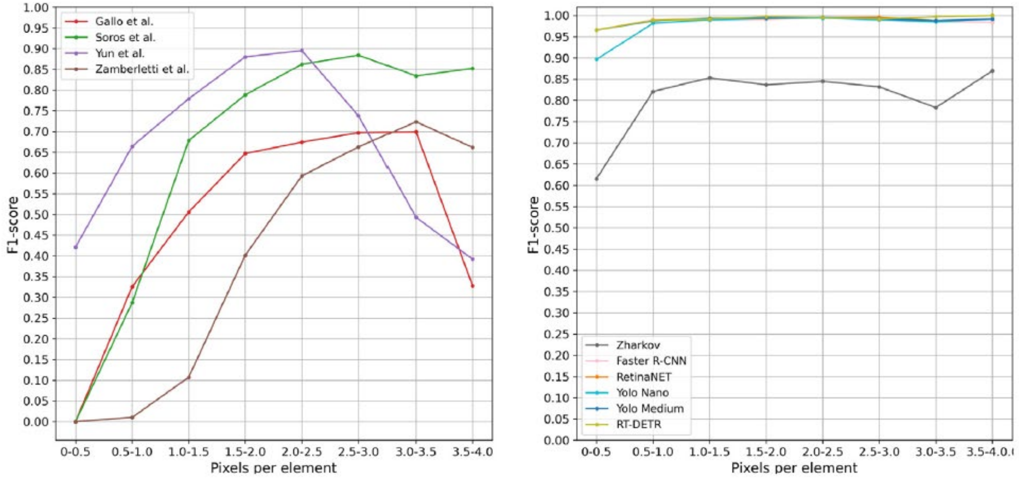
Figure 5. Loss values computed on the test set of CIFAR-10 using different methods.


(a) Ground-Truth
(b) nnU-Net ResEnc
(c) nnFormer
(d) UNETR++
(e) UMamba
(f) VMamba
(g) Swin-UMamba

BarBeR

BARCODE BENCHMARK REPOSITORY

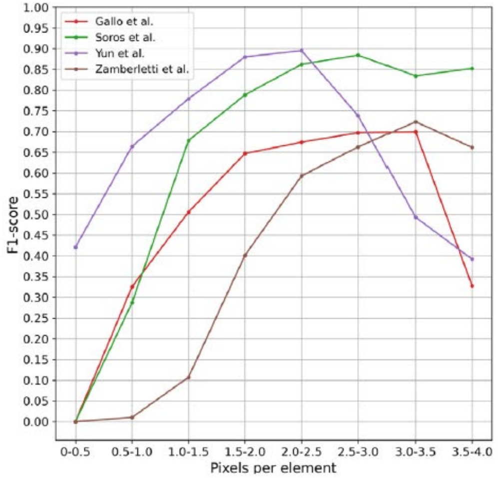
Linear barcodes at different ranges of pixels per element. Deep vs non-deep solutions.

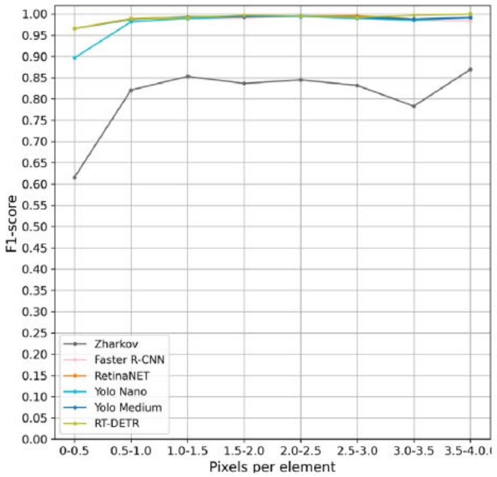


BarBeR

BARCODE BENCHMARK REPOSITORY

Linear barcodes at different ranges of pixels per element. Deep vs non-deep solutions.





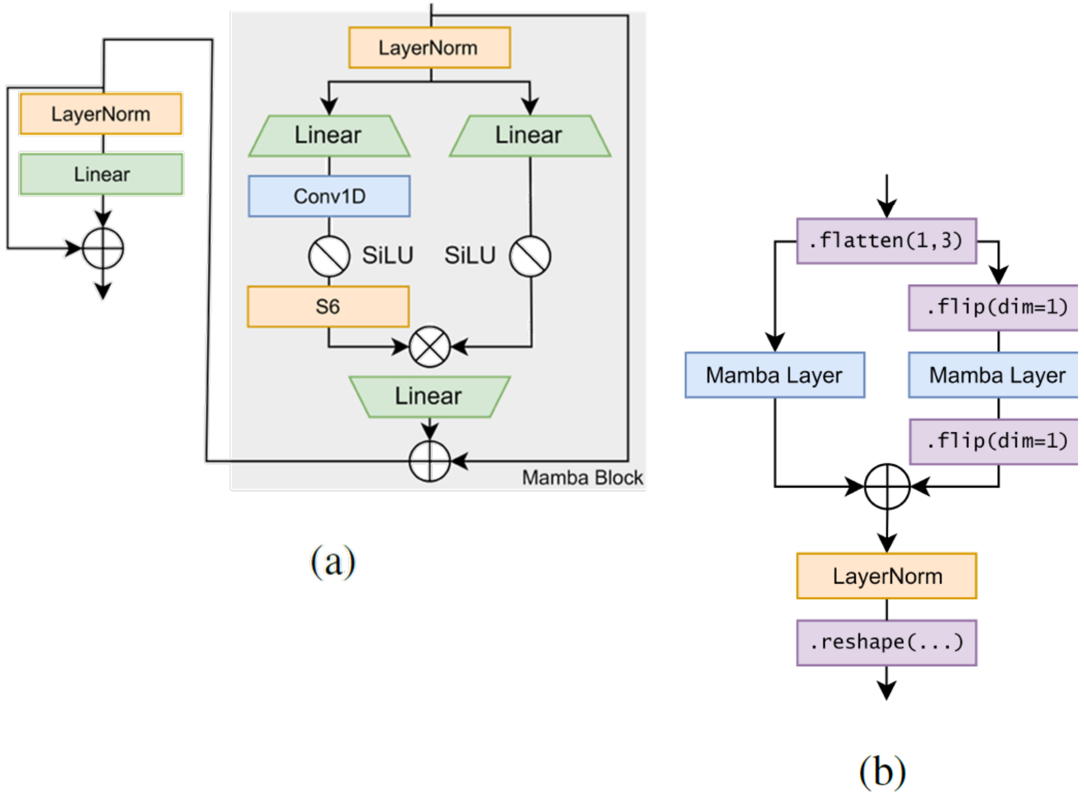


Fig. 1: (a) depicts our (unidirectional) Mamba Layer (in gray the original Mamba block), while (b) represents our bidirectional 3D Mamba layer, which includes (a).

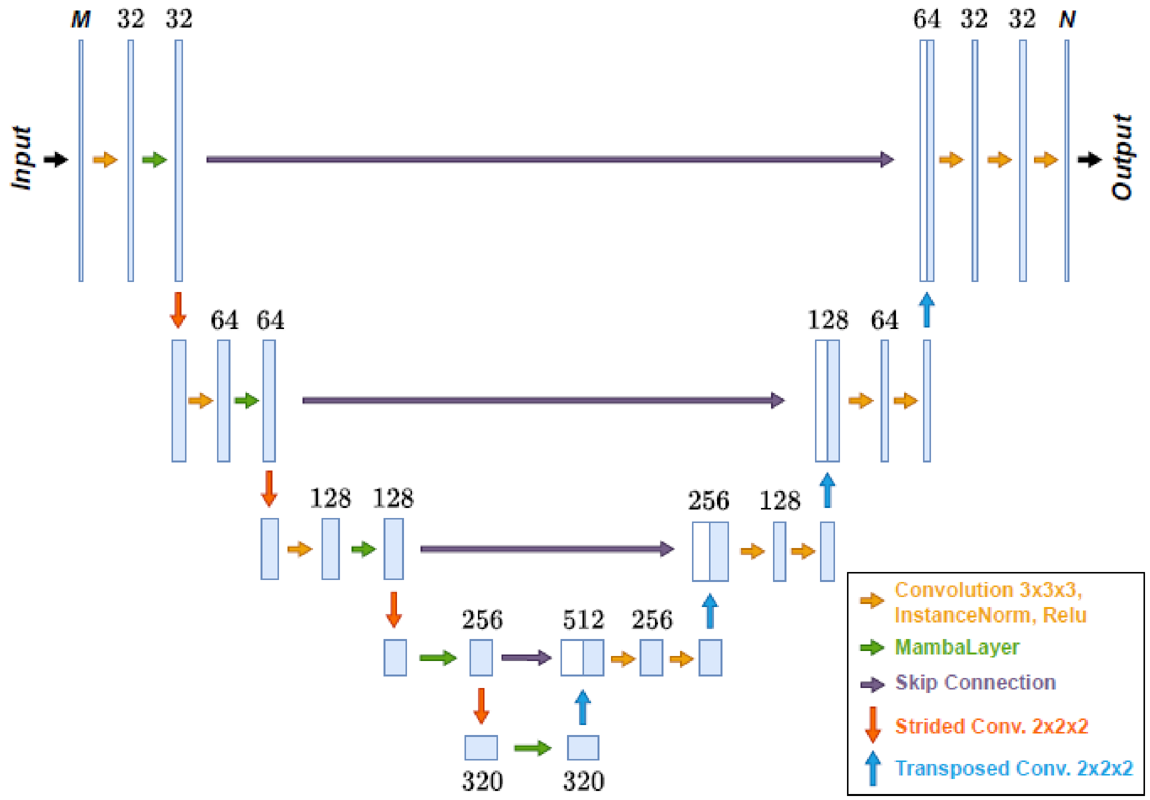


Fig. 2: U-Net-like architecture integrating our proposed Mamba layers. By properly selecting the Mamba layers (turquoise arrows), *Unidirectional*, *Bidirectional*, and *Multi-directional* are obtained.

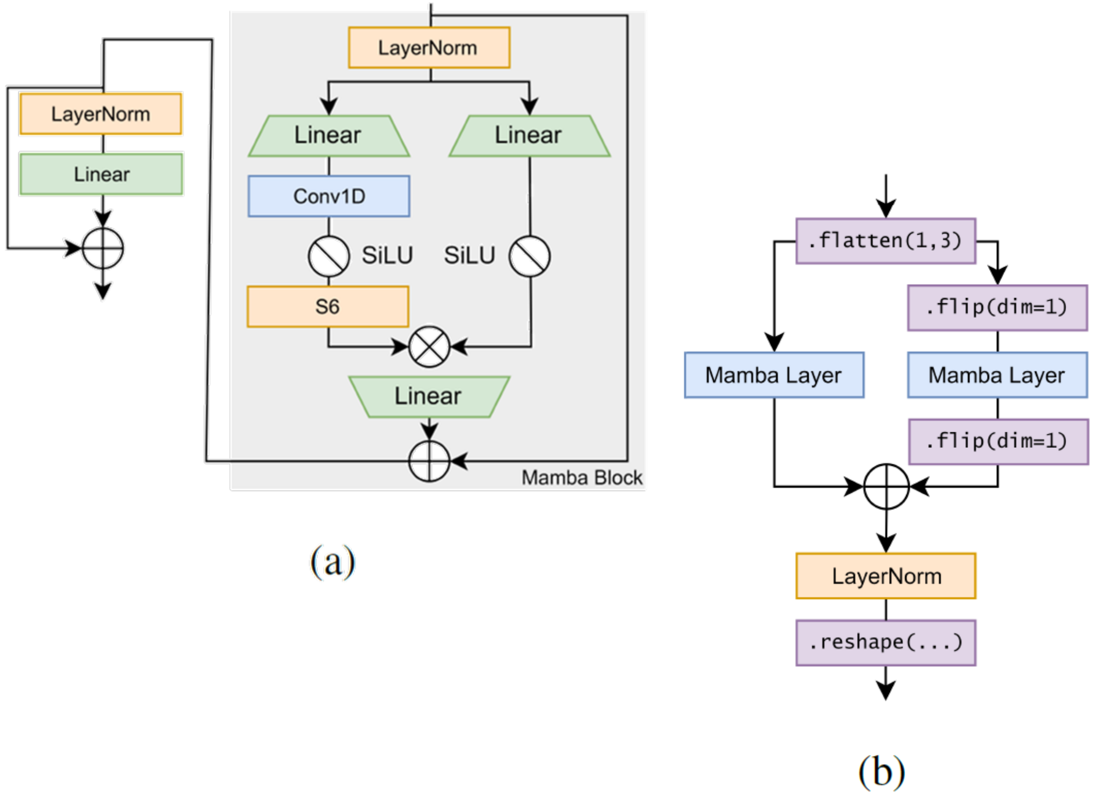


Fig. 1: (a) depicts our (unidirectional) Mamba Layer (in gray the original Mamba block), while (b) represents our bidirectional 3D Mamba layer, which includes (a).

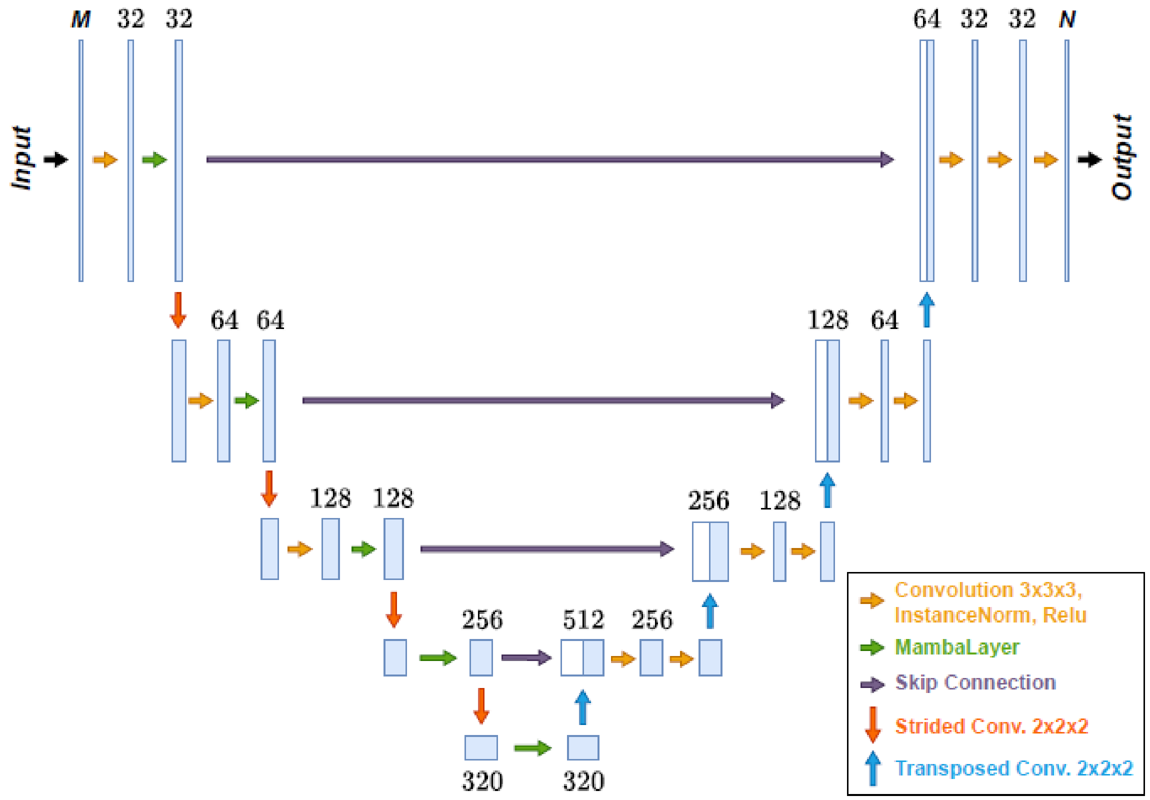
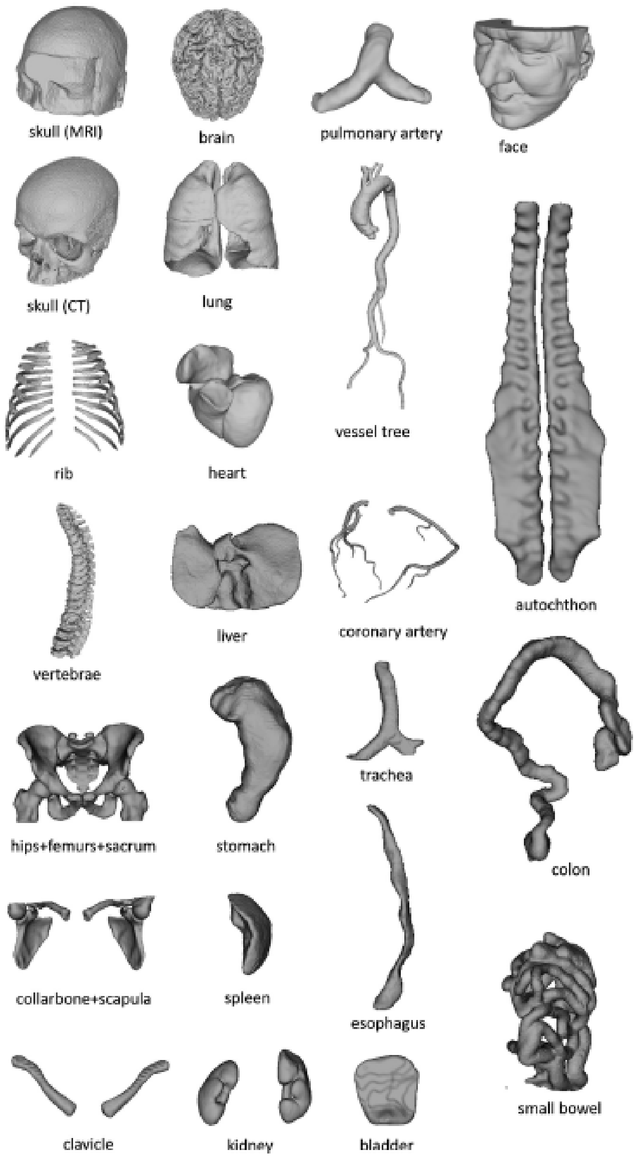
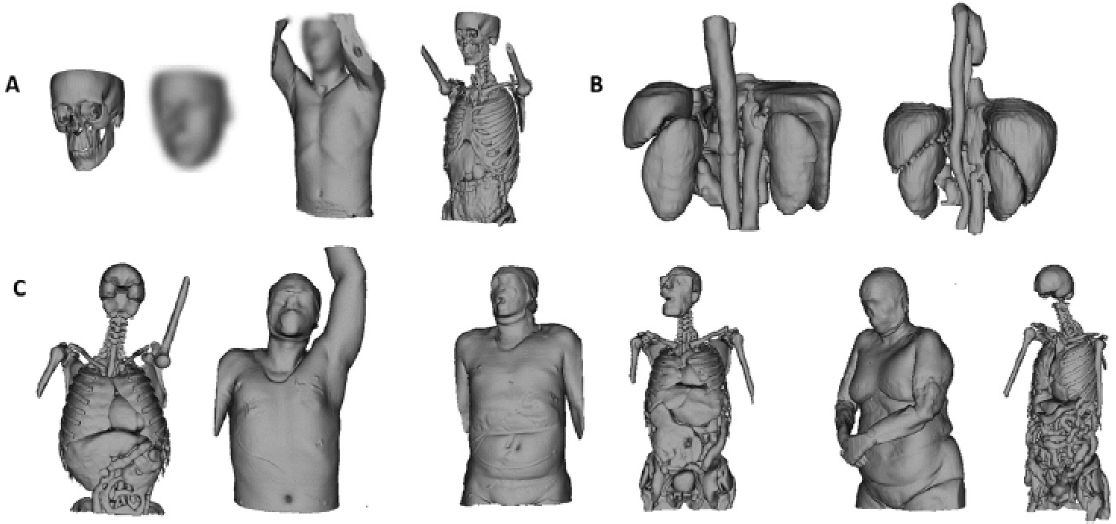
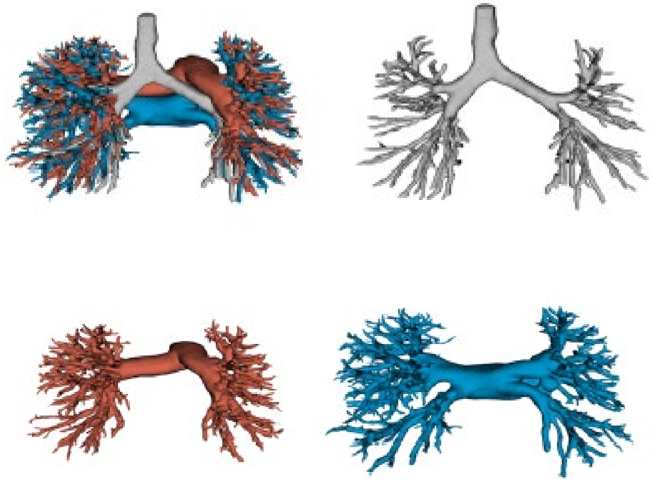


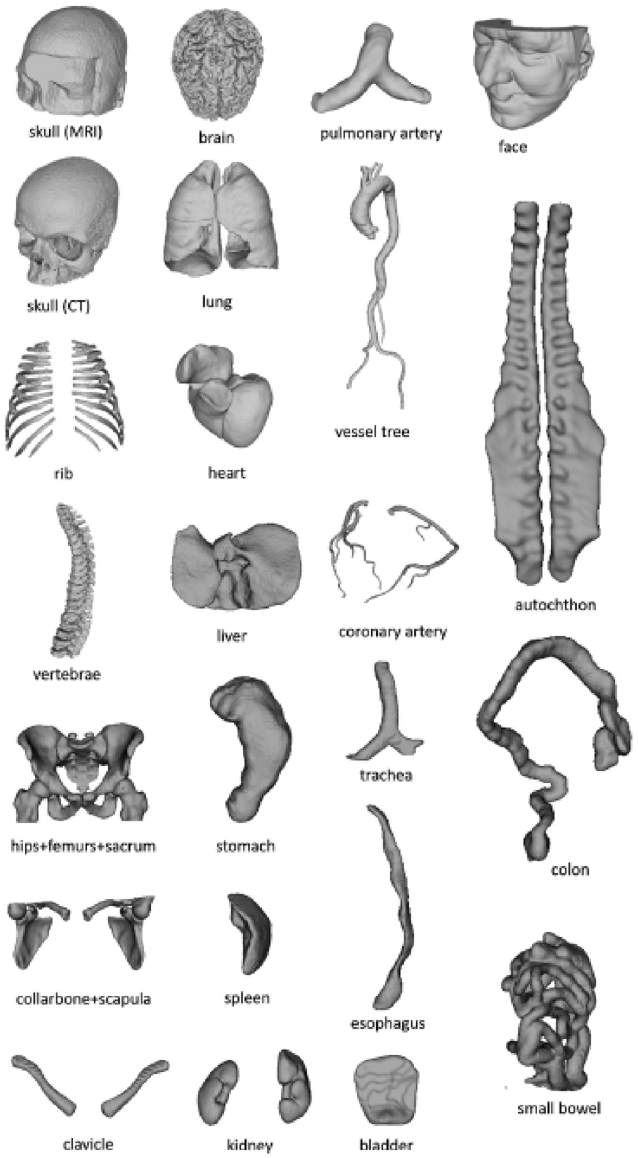
Fig. 2: U-Net-like architecture integrating our proposed Mamba layers. By properly selecting the Mamba layers (turquoise arrows), *Unidirectional*, *Bidirectional*, and *Multi-directional* are obtained.


skull (MRI)
brain
pulmonary artery
face
skull (CT)
lung
vessel tree
autochthon
rib
heart
vertebrae
liver
coronary artery
hips+femurs+sacrum
stomach
trachea
colon
collarbone+scapula
spleen
esophagus
small bowel
clavicle
kidney
bladder

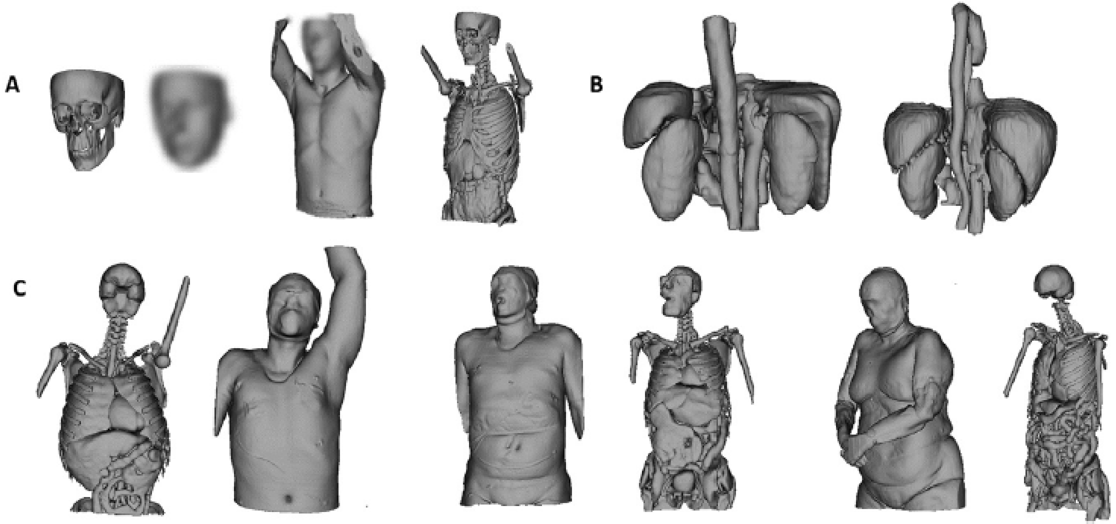

A
B
C
D
E
F

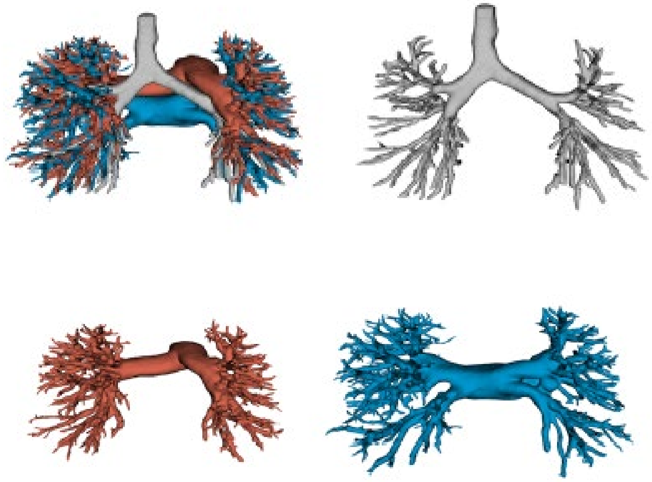

A
B
C

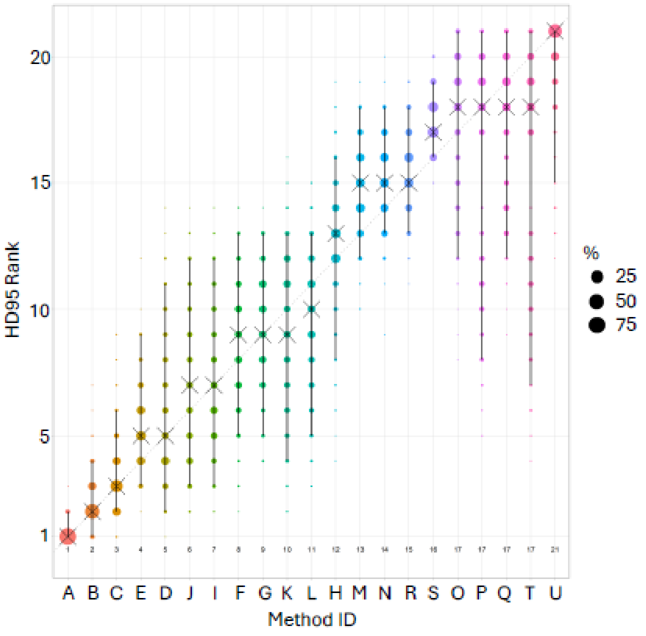


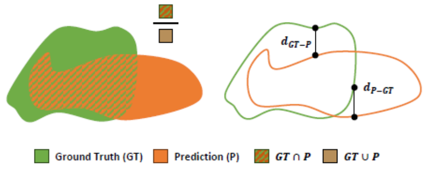

skull (MRI)
brain
pulmonary artery
face
skull (CT)
lung
vessel tree
rib
heart
autochthon
vertebrae
liver
coronary artery
trachea
hips+femurs+sacrum
stomach
colon
collarbone+scapula
spleen
esophagus
small bowel
clavicle
kidney
bladder

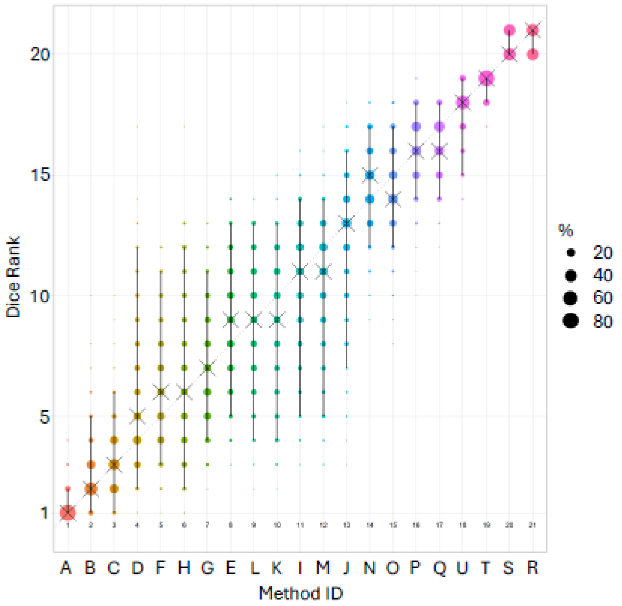

A
B
C
D
E
F


A
B
C



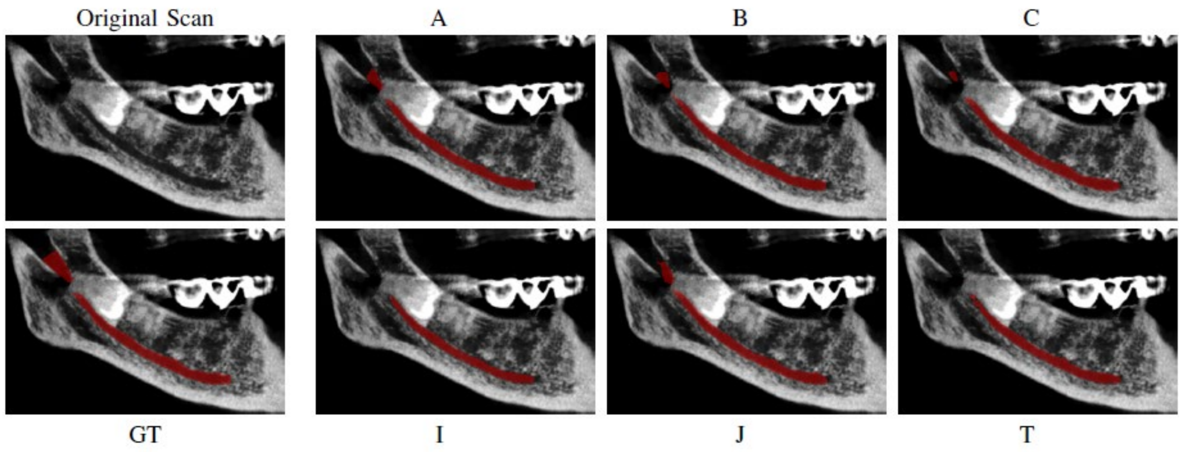


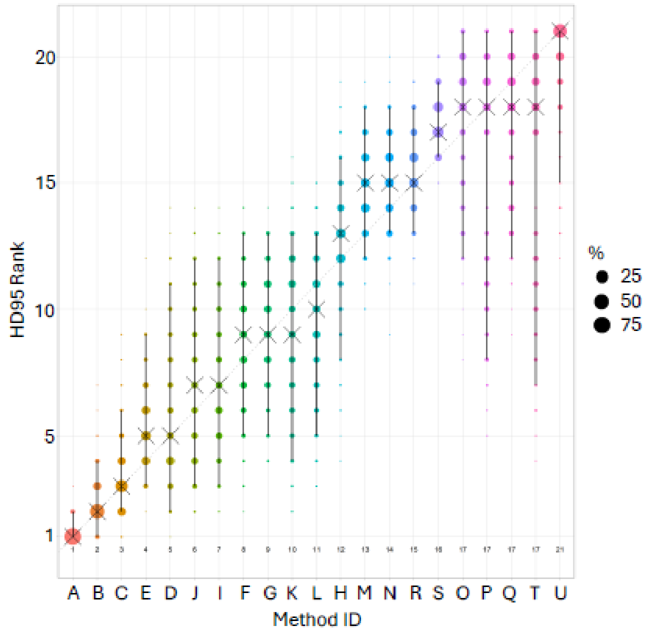


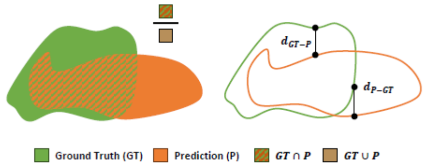


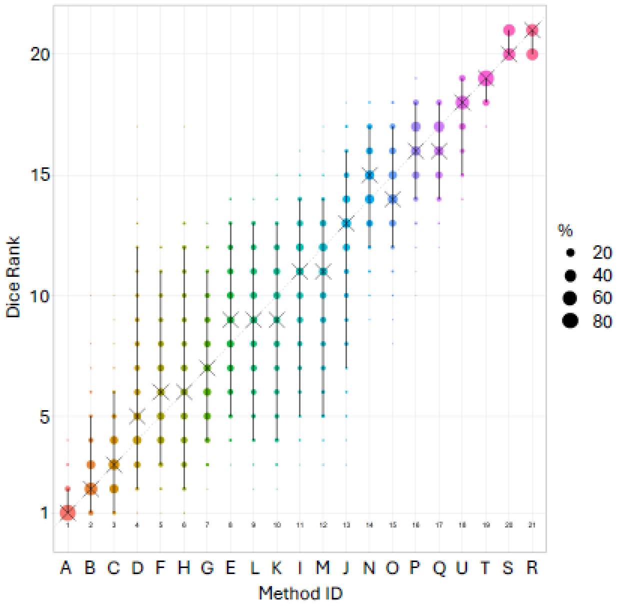
TOOTH FAIRY

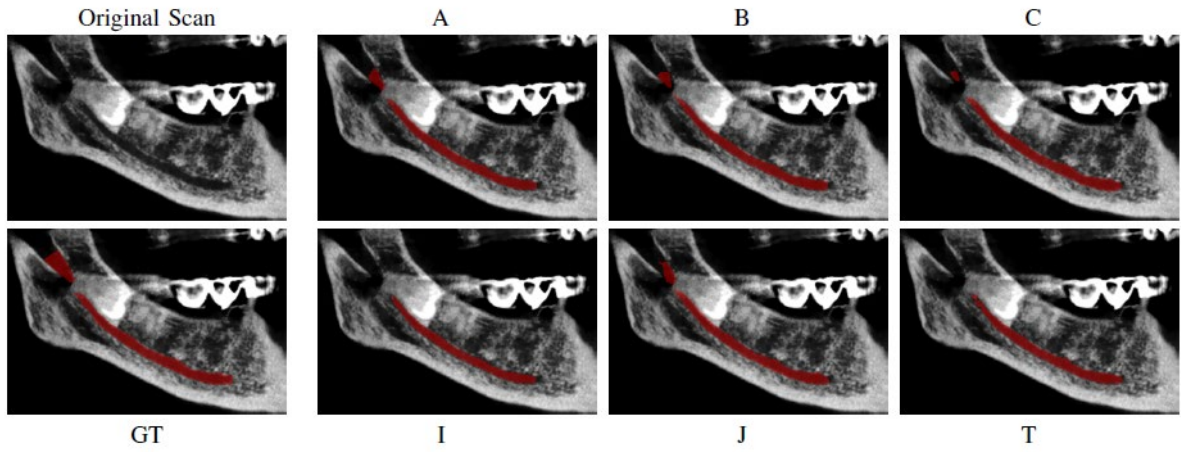
Final Rank	ID	Participant	Country	Dice		HD95	
				Score	Rank	Score	Rank
1	A	Liu et al.	China	0.796 ± 0.093	1	4.49 ± 6.08	1
2	B	Wang et al.	China	0.769 ± 0.106	2	7.45±10.90	2
3	C	Wodzinski et al.	Switzerland	0.760 ± 0.088	3	9.22±14.40	3
4	D	Kirchoff et al.	Germany	0.739 ± 0.146	4	13.90±28.70	5
5	E	Huang	China	0.715 ± 0.110	8	13.10±13.50	4
6	F	Su	China	0.734 ± 0.099	5	18.00±25.30	8
7	G	Ye	China	0.728 ± 0.094	7	18.90±25.00	9
8	H	Yang	China	0.731 ± 0.151	6	28.60±47.40	12
8	I	Han et al.	South Korea	0.700 ± 0.148	11	15.70±24.80	7
10	J	Szczepański et al.	Poland	0.675 ± 0.208	13	14.90±29.30	6
11	K	Wu	China	0.712 ± 0.142	10	19.40±35.60	10
11	L	Zheng	China	0.713 ± 0.129	9	19.50±28.00	11
*13	M	Lumetti et al.	Italy	0.699 ± 0.151	12	38.6 ±48.0	13
14	N	Han et al.	South Korea	0.643 ± 0.179	14	40.20±49.70	14
15	O	Huang	China	0.642 ± 0.187	15	inf	17
16	P	Szczepański et al.	Poland	0.613 ± 0.201	16	inf	17
17	Q	Pang	China	0.612 ± 0.178	17	inf	17
18	R	Gamal	Egypt	0.326 ± 0.172	21	42.70±43.90	15
18	S	Li	China	0.327 ± 0.218	20	65.90±48.10	16
18	T	Caselles Ballester et al.	Spain	0.507 ± 0.209	19	inf	17
21	U	Kirchoff et al.	Germany	0.565 ± 0.259	18	inf	21



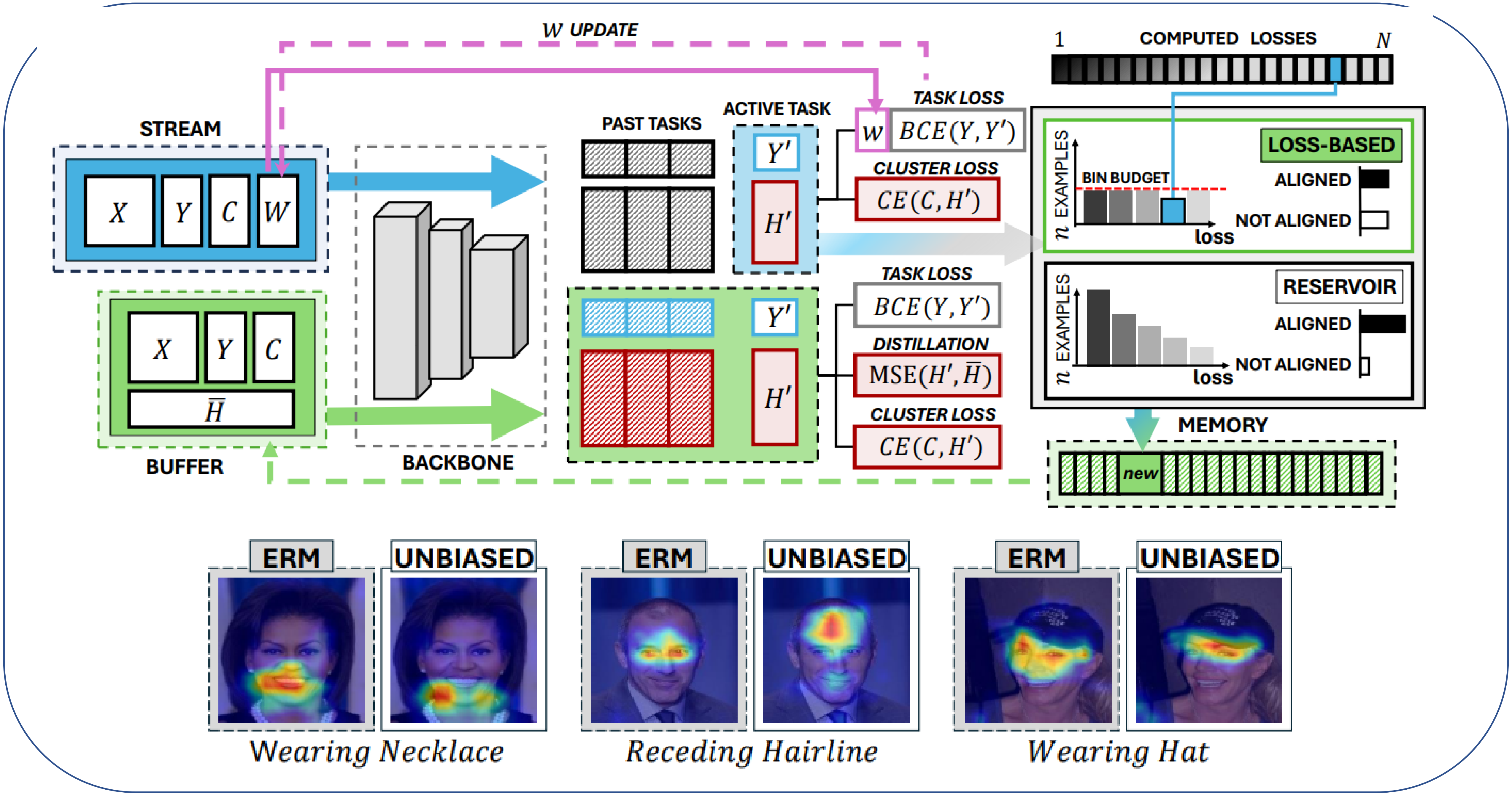




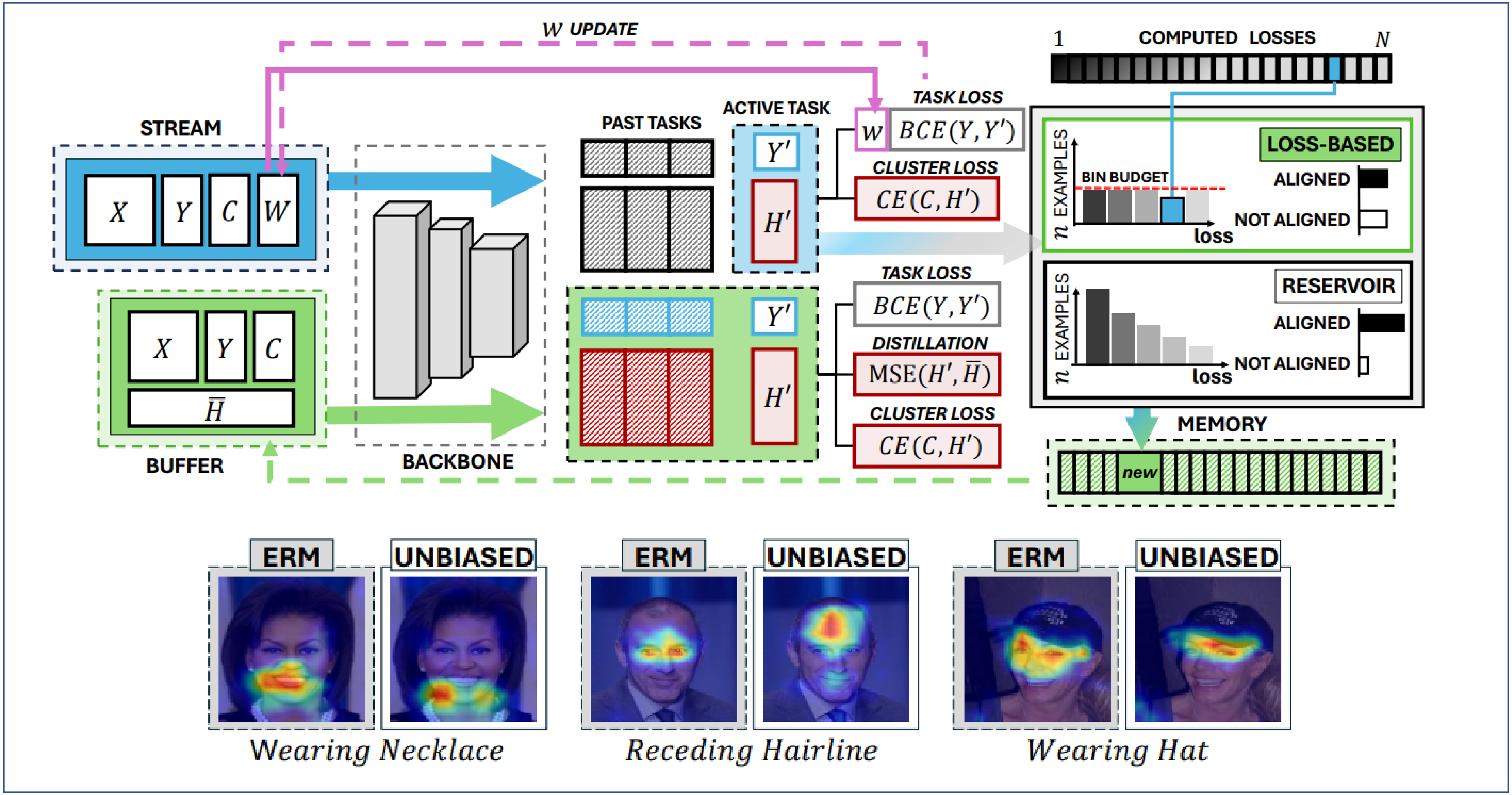




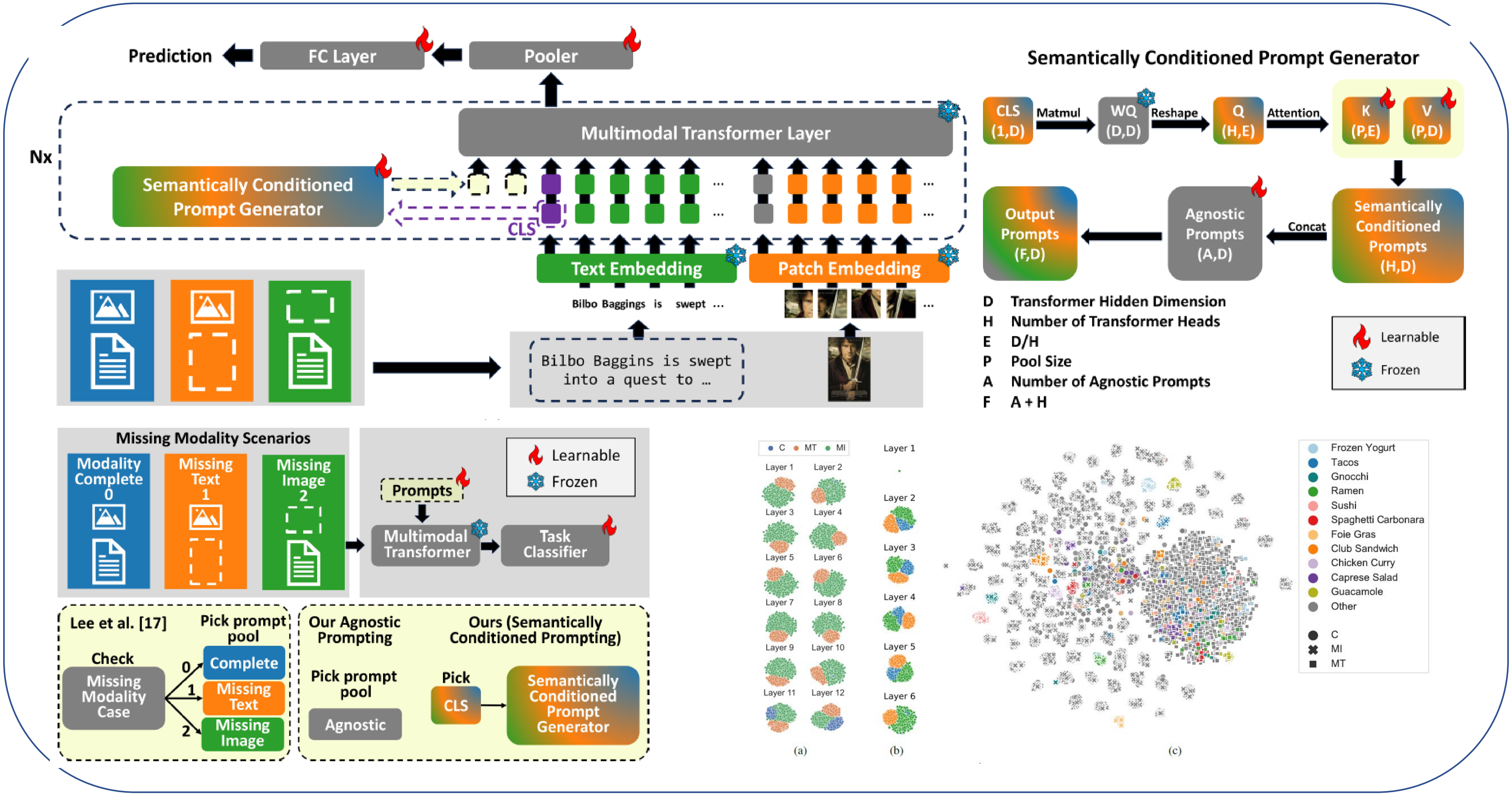
Final Rank	ID	Participant	Country	Dice		HD95	
				Score	Rank	Score	Rank
1	A	Liu et al.	China	0.796 ± 0.093	1	4.49 ± 6.08	1
2	B	Wang et al.	China	0.769 ± 0.106	2	7.45 ± 10.90	2
3	C	Wodzinski et al.	Switzerland	0.760 ± 0.088	3	9.22 ± 14.40	3
4	D	Kirchoff et al.	Germany	0.739 ± 0.146	4	13.90 ± 28.70	5
5	E	Huang	China	0.715 ± 0.110	8	13.10 ± 13.50	4
6	F	Su	China	0.734 ± 0.099	5	18.00 ± 25.30	8
7	G	Ye	China	0.728 ± 0.094	7	18.90 ± 25.00	9
8	H	Yang	China	0.731 ± 0.151	6	28.60 ± 47.40	12
8	I	Han et al.	South Korea	0.700 ± 0.148	11	15.70 ± 24.80	7
10	J	Szczepański et al.	Poland	0.675 ± 0.208	13	14.90 ± 29.30	6
11	K	Wu	China	0.712 ± 0.142	10	19.40 ± 35.60	10
11	L	Zheng	China	0.713 ± 0.129	9	19.50 ± 28.00	11
*13	M	Lumetti et al.	Italy	0.699 ± 0.151	12	38.6 ± 48.0	13
14	N	Han et al.	South Korea	0.643 ± 0.179	14	40.20 ± 49.70	14
15	O	Huang	China	0.642 ± 0.187	15	inf	17
16	P	Szczepański et al.	Poland	0.613 ± 0.201	16	inf	17
17	Q	Pang	China	0.612 ± 0.178	17	inf	17
18	R	Gamal	Egypt	0.326 ± 0.172	21	42.70 ± 43.90	15
18	S	Li	China	0.327 ± 0.218	20	65.90 ± 48.10	16
18	T	Caselles Ballester et al.	Spain	0.507 ± 0.209	19	inf	17
21	U	Kirchoff et al.	Germany	0.565 ± 0.259	18	inf	21

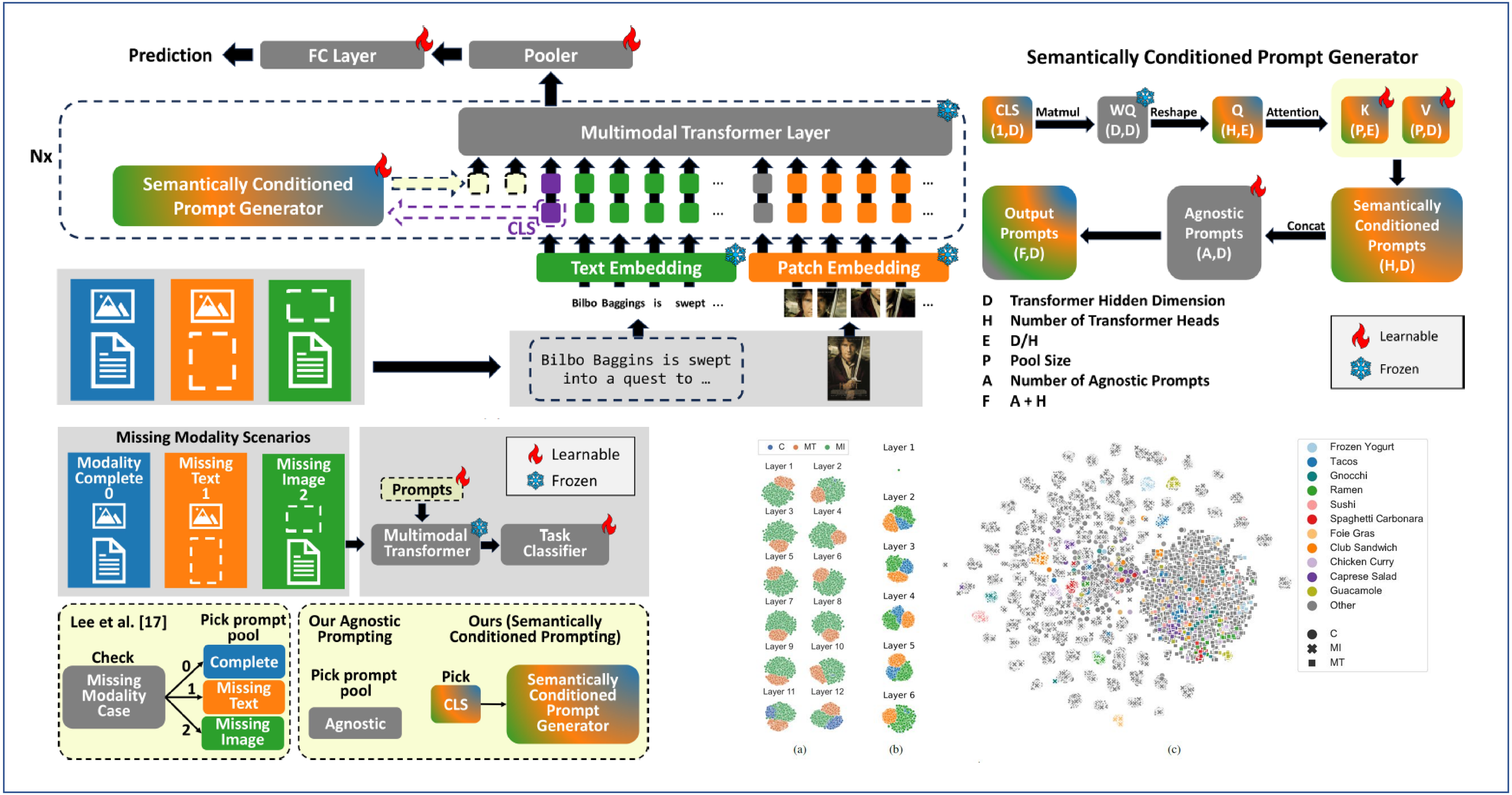


W UPDATE
1
COMPUTED LOSSES
N
STREAM
X
Y
C
W
PAST TASKS
ACTIVE TASK
Y′
H′
TASK LOSS
W
BCE(Y,Y′)
CLUSTER LOSS
CE(C,H′)
LOSS-BASED
n EXAMPLES
BIN BUDGET
ALIGNED
NOT ALIGNED
loss
RESERVOIR
n EXAMPLES
ALIGNED
NOT ALIGNED
loss
BUFFER
X
Y
C
H̄
BACKBONE
Y′
H′
TASK LOSS
BCE(Y,Y′)
DISTILLATION
MSE(H′,H̄)
CLUSTER LOSS
CE(C,H′)
MEMORY
new
ERM
UNBIASED
Wearing Necklace
ERM
UNBIASED
Receding Hairline
ERM
UNBIASED
Wearing Hat



W UPDATE
COMPUTED LOSSES
1
N
STREAM
X
Y
C
W
PAST TASKS
ACTIVE TASK
Y'
H'
TASK LOSS
W
BCE(Y, Y')
CLUSTER LOSS
CE(C, H')
LOSS-BASED
n EXAMPLES
BIN BUDGET
loss
ALIGNED
NOT ALIGNED
RESERVOIR
n EXAMPLES
loss
ALIGNED
NOT ALIGNED
BUFFER
X
Y
C
H̄
BACKBONE
Y'
H'
TASK LOSS
BCE(Y, Y')
DISTILLATION
MSE(H', H̄)
CLUSTER LOSS
CE(C, H')
MEMORY
new
ERM
UNBIASED
Wearing Necklace
ERM
UNBIASED
Receding Hairline
ERM
UNBIASED
Wearing Hat


Prediction
FC Layer
Pooler
Nx
Multimodal Transformer Layer
Semantically Conditioned Prompt Generator
CLS
Text Embedding
Patch Embedding
Bilbo Baggings is swept ...
Bilbo Baggins is swept into a quest to ...
Semantically Conditioned Prompt Generator
CLS (1,D)
Matmul
WQ (D,D)
Reshape
Q (H,E)
Attention
K (P,E)
V (P,D)
Output Prompts (F,D)
Agnostic Prompts (A,D)
Concat
Semantically Conditioned Prompts (H,D)
D Transformer Hidden Dimension
H Number of Transformer Heads
E D/H
P Pool Size
A Number of Agnostic Prompts
F A + H
Learnable
Frozen
Missing Modality Scenarios
Modality Complete 0
Missing Text 1
Missing Image 2
Prompts
Multimodal Transformer
Task Classifier
Learnable
Frozen
Lee et al. [17]
Pick prompt pool
Check
Missing Modality Case
0
1
2
Complete
Missing Text
Missing Image
Our Agnostic Prompting
Pick prompt pool
Agnostic
Ours (Semantically Conditioned Prompting)
Pick
CLS
Semantically Conditioned Prompt Generator
C MT MI
Layer 1 Layer 2 Layer 3 Layer 4 Layer 5 Layer 6 Layer 7 Layer 8 Layer 9 Layer 10 Layer 11 Layer 12
(a)
Layer 1 Layer 2 Layer 3 Layer 4 Layer 5 Layer 6
(b)
Frozen Yogurt
Tacos
Gnocchi
Ramen
Sushi
Spaghetti Carbonara
Foie Gras
Club Sandwich
Chicken Curry
Caprese Salad
Guacamole
Other
C
MI
MT
(c)

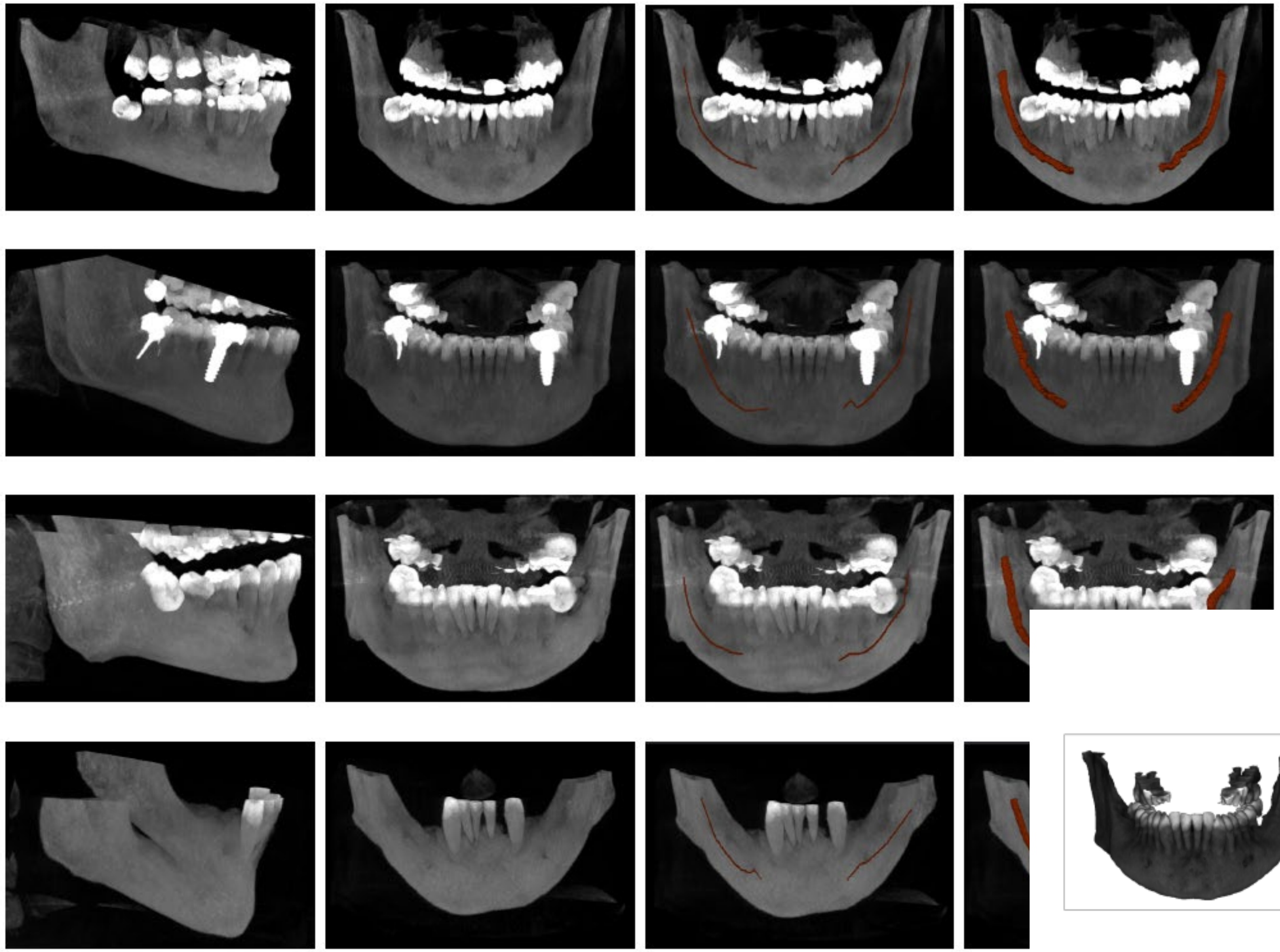

Prediction
FC Layer
Pooler
Multimodal Transformer Layer
Nx
Semantically Conditioned Prompt Generator
CLS
Text Embedding
Patch Embedding
Bilbo Baggings is swept ...
Bilbo Baggins is swept into a quest to ...
Semantically Conditioned Prompt Generator
CLS (1,D)
Matmul
WQ (D,D)
Reshape
Q (H,E)
Attention
K (P,E)
V (P,D)
Semantically Conditioned Prompts (H,D)
Concat
Agnostic Prompts (A,D)
Output Prompts (F,D)
D Transformer Hidden Dimension
H Number of Transformer Heads
E D/H
P Pool Size
A Number of Agnostic Prompts
F A + H
Learnable
Frozen
Missing Modality Scenarios
Modality Complete 0
Missing Text 1
Missing Image 2
Prompts
Multimodal Transformer
Task Classifier
Learnable
Frozen
Lee et al. [17]
Check
Missing Modality Case
Pick prompt pool
Complete
Missing Text
Missing Image
Our Agnostic Prompting
Pick prompt pool
Agnostic
Ours (Semantically Conditioned Prompting)
Pick
CLS
Semantically Conditioned Prompt Generator
C MT MI
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 7
Layer 8
Layer 9
Layer 10
Layer 11
Layer 12
(a)
(b)
(c)
Frozen Yogurt
Tacos
Gnocchi
Ramen
Sushi
Spaghetti Carbonara
Foie Gras
Club Sandwich
Chicken Curry
Caprese Salad
Guacamole
Other
C
MI
MT

MICCAI 2024
Marrakesh
MOROCCO

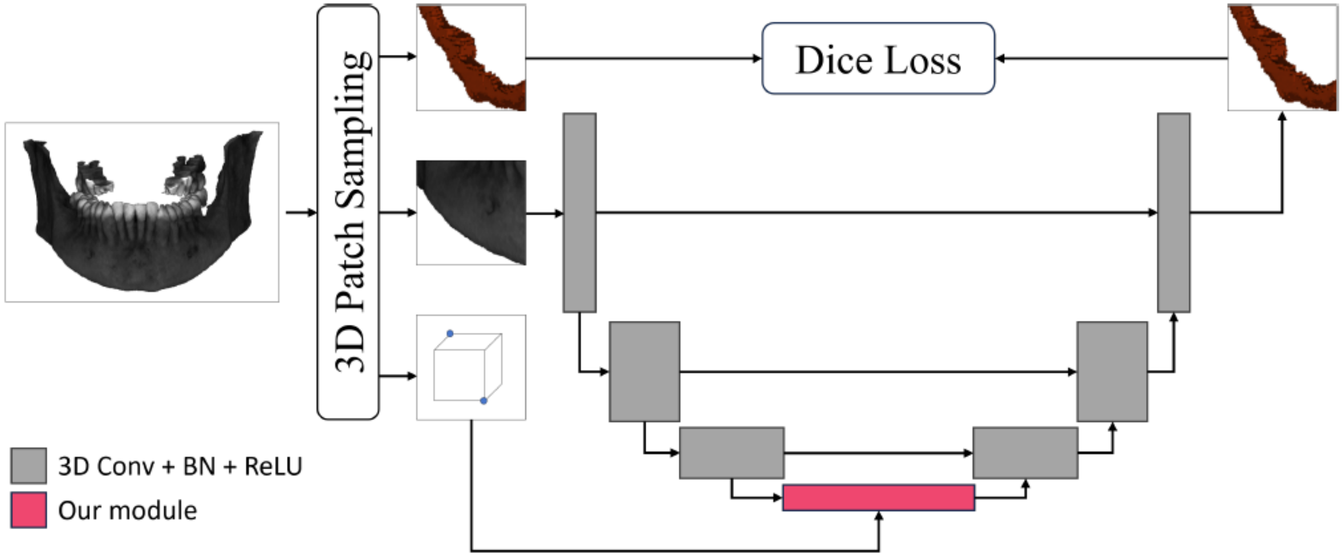
TOOTH FAIRY2

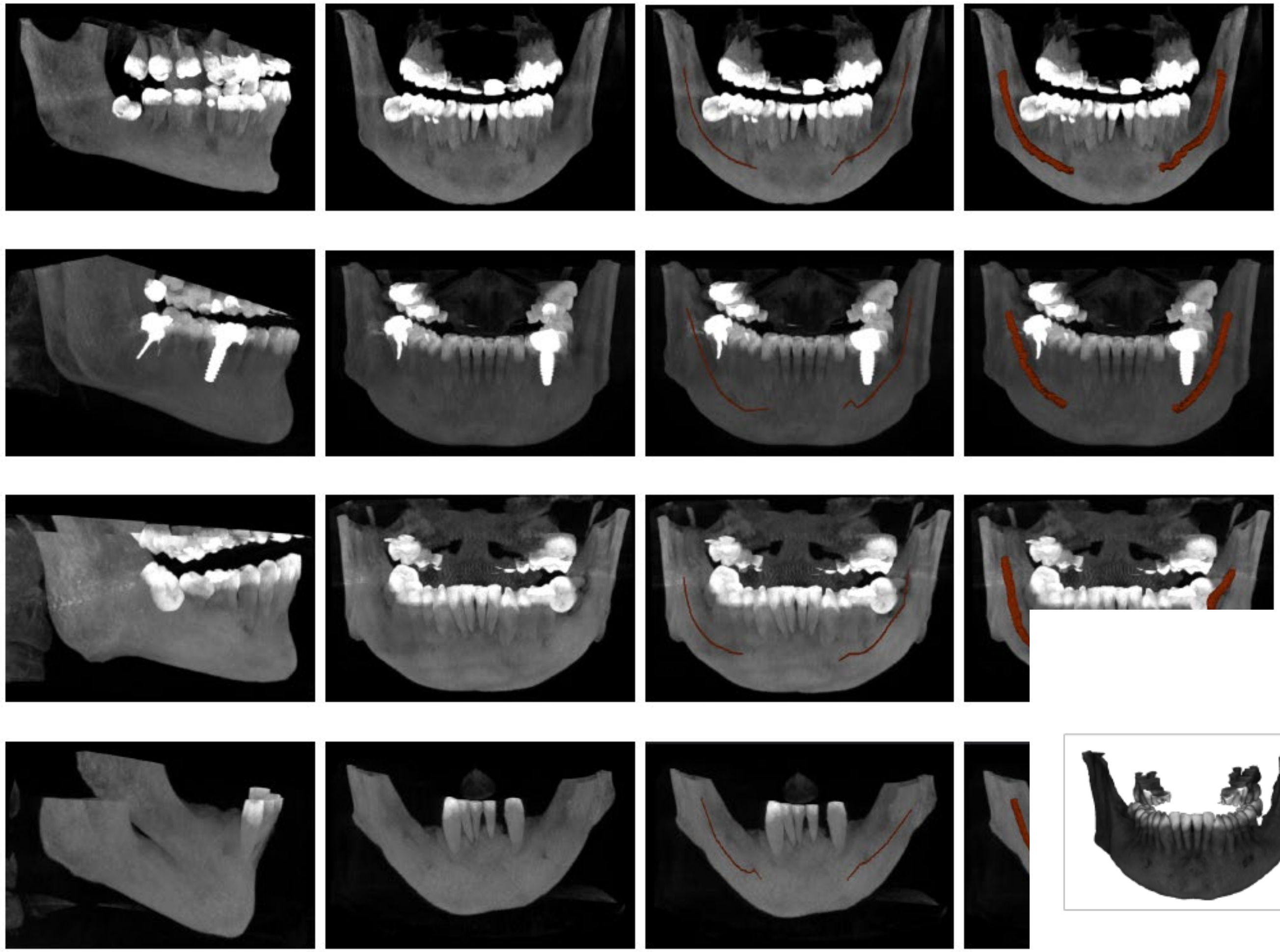
MICCAI 2024
Marrakesh
MOROCCO

TOOTH
FAIRY2

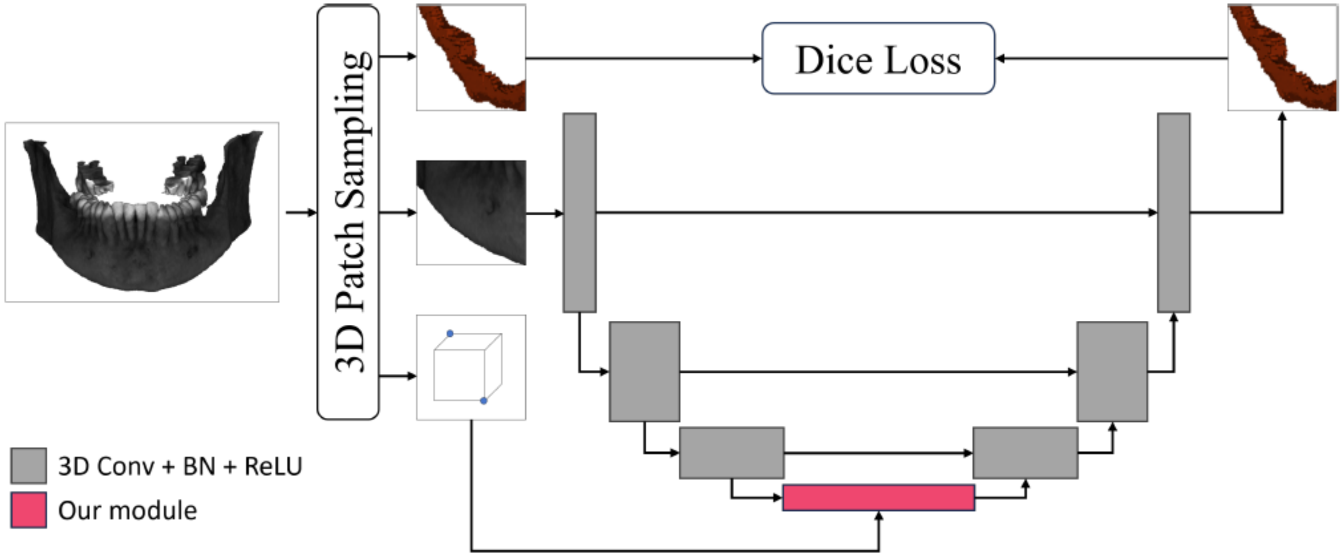


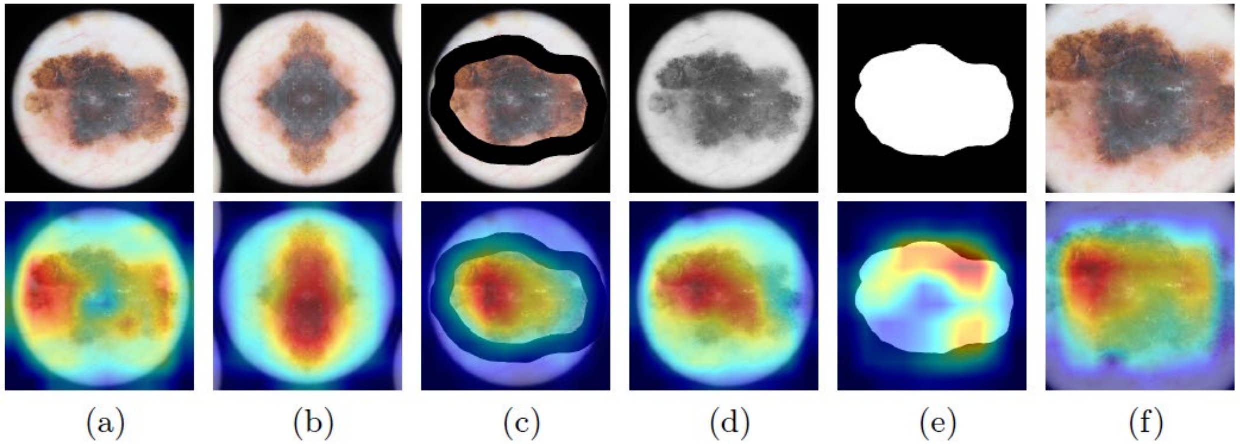
Dataset	Method	IoU	Dice
Maxillo	Usman *et al.* [23]	–	0.770
	Cripriano *et al.* [6]	0.650	0.790
	Zhao *et al.* [27]	–	0.810
	Ours	**0.704**	**0.824**
ToothFairy	Ours	**0.710**	**0.831**



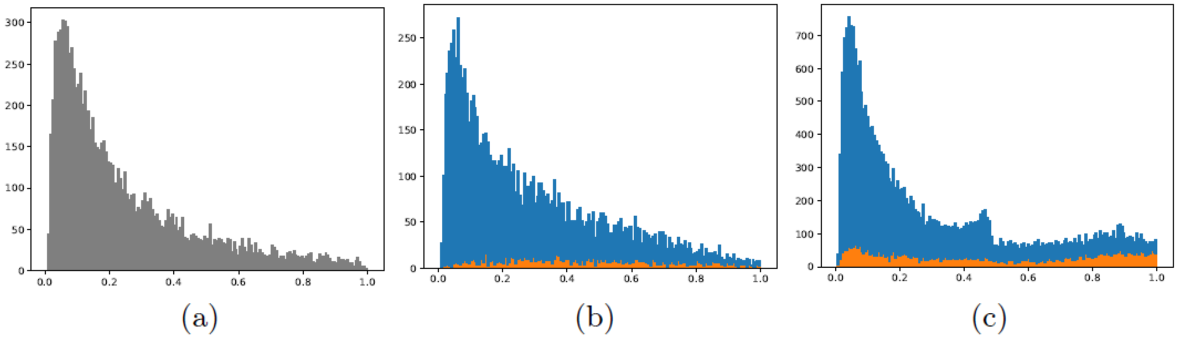


Dataset	Method	IoU	Dice
Maxillo	Usman *et al.* [23]	–	0.770
	Cripriano *et al.* [6]	0.650	0.790
	Zhao *et al.* [27]	–	0.810
	Ours	**0.704**	**0.824**
ToothFairy	Ours	**0.710**	**0.831**

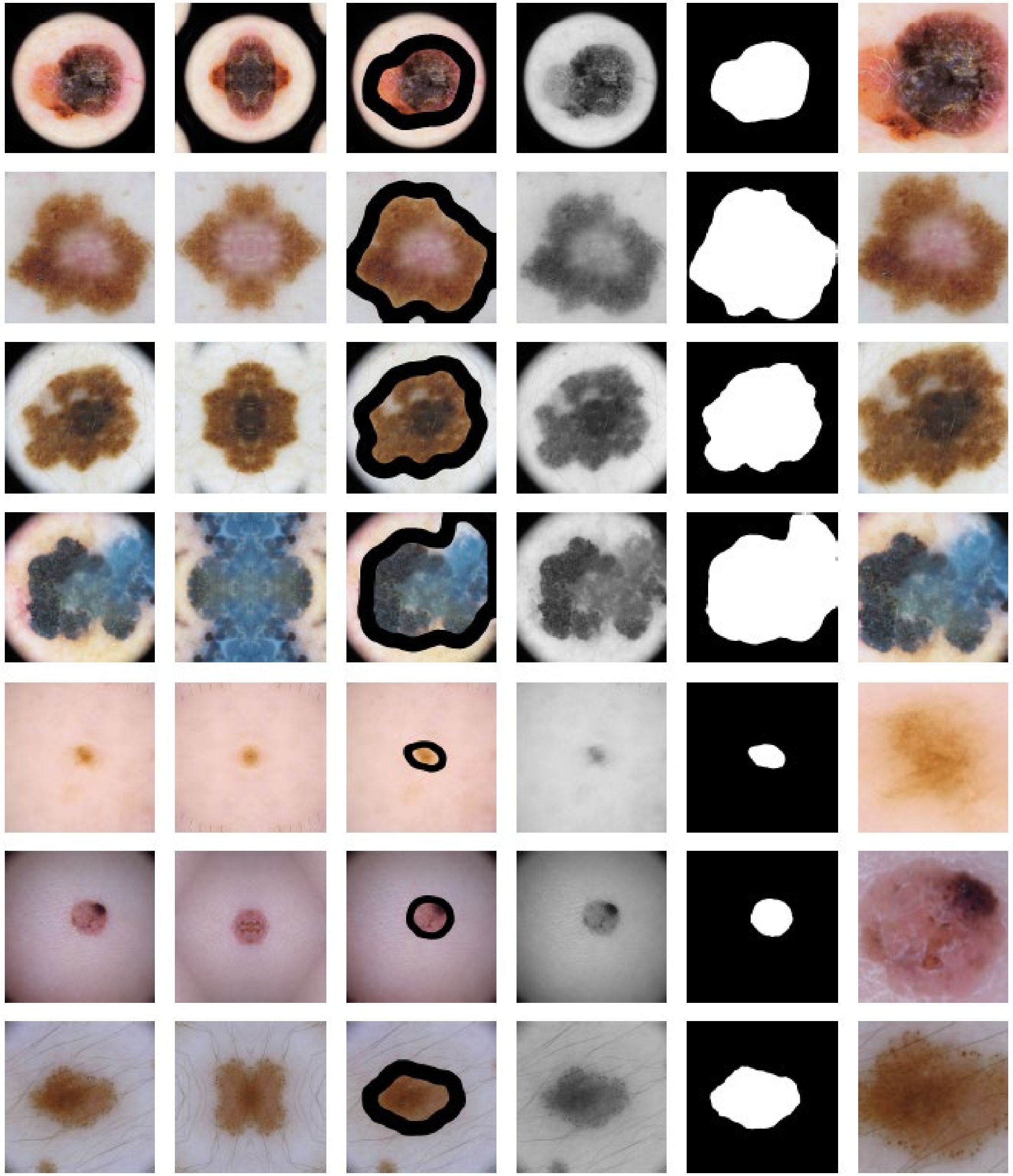


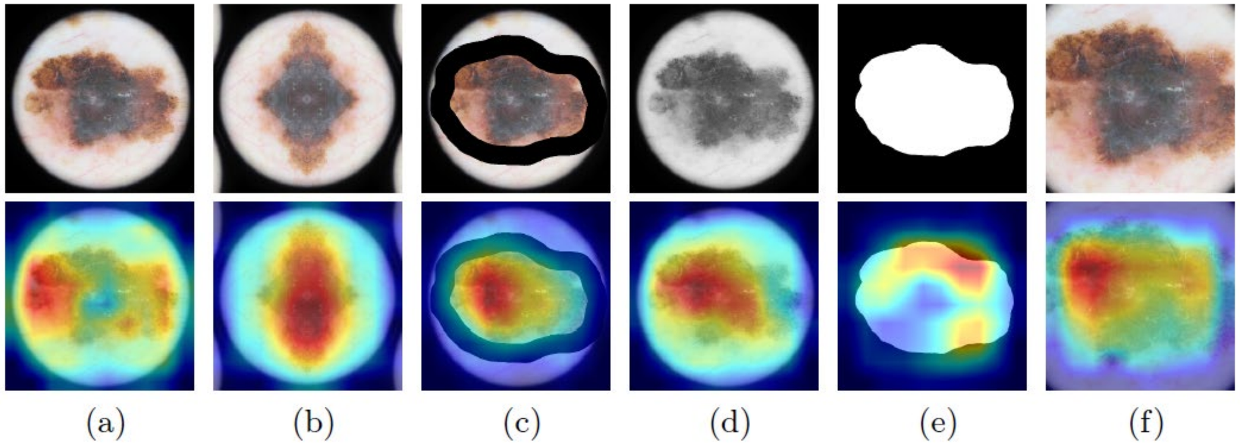


Grad-CAM visualization when debasing different ABCD(E) properties. (a) Original, (b) Asymmetry, (c) Borders, (d) Grayscale, (e) Mask, (f) Diameter.

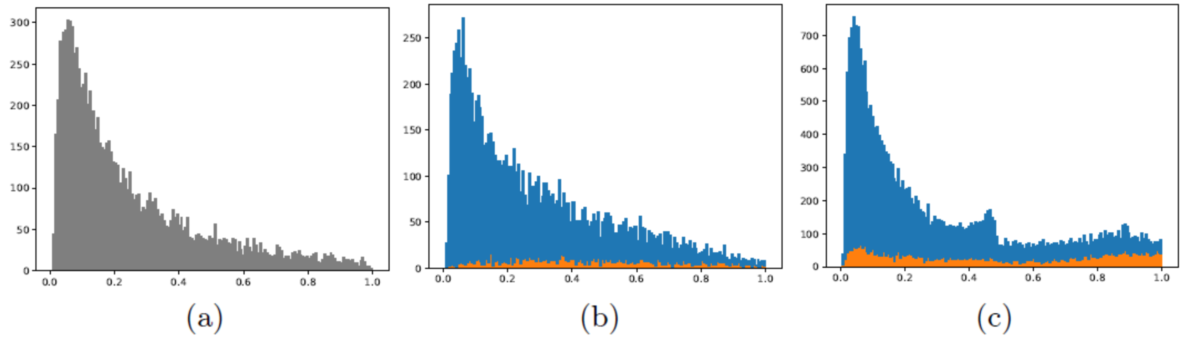


Histograms of foreground density distribution within different test sets. (a) ISIC2020 official test set, (b) ISIC19-20 “Internal” test set, (c) Private Dataset. Benign and malignant skin lesions are depicted in blue and orange respectively.

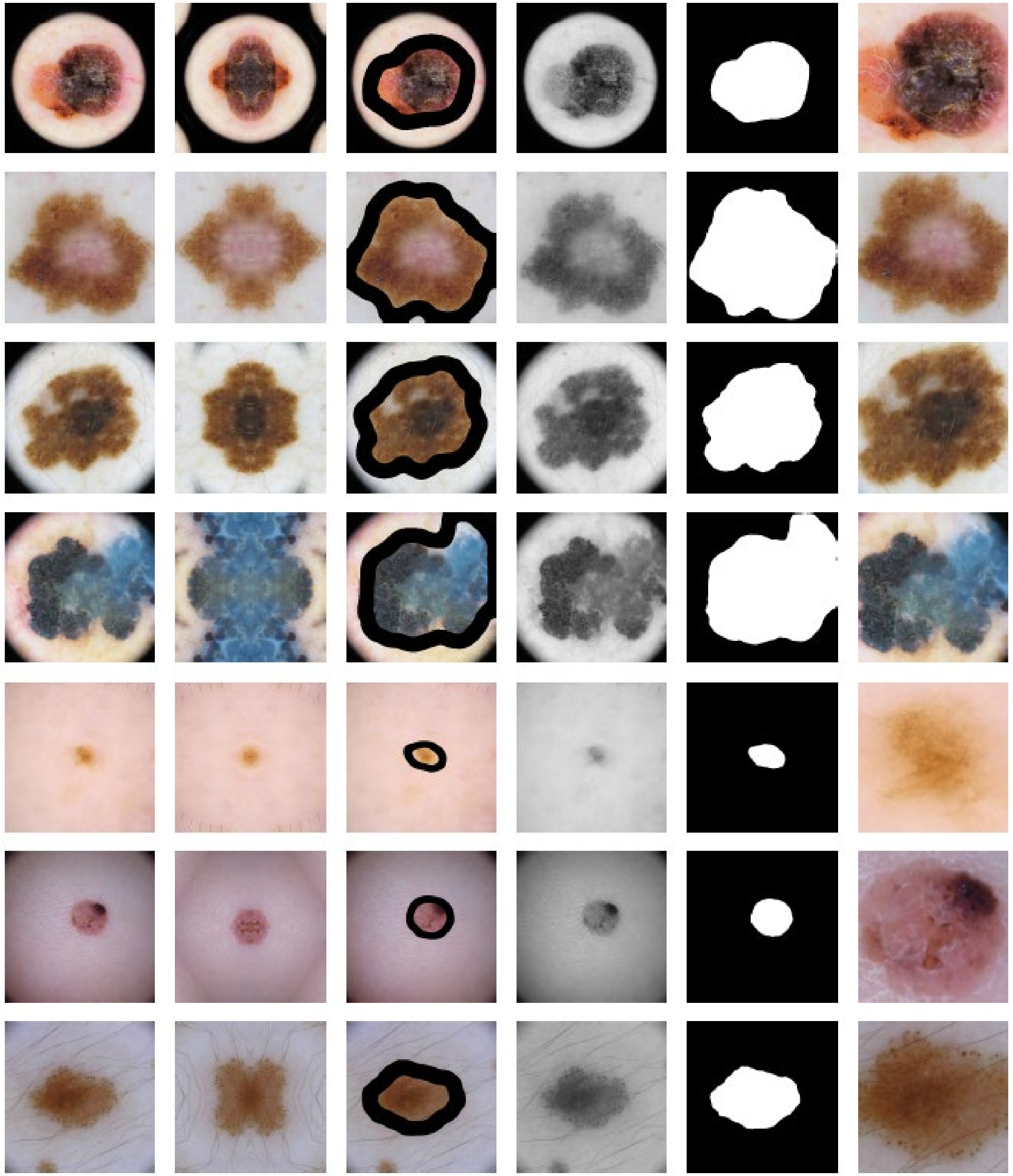


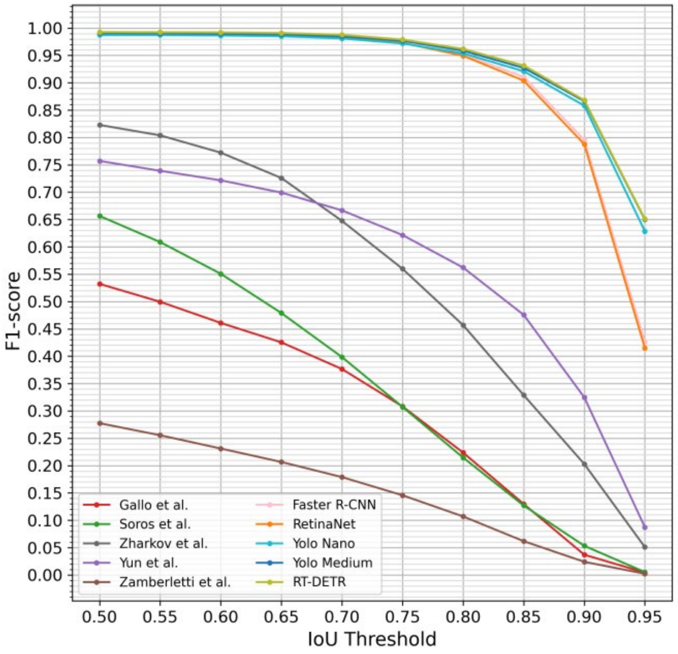


Grad-CAM visualization when debasing different ABCD(E) properties. (a) Original, (b) Asymmetry, (c) Borders, (d) Grayscale, (e) Mask, (f) Diameter.

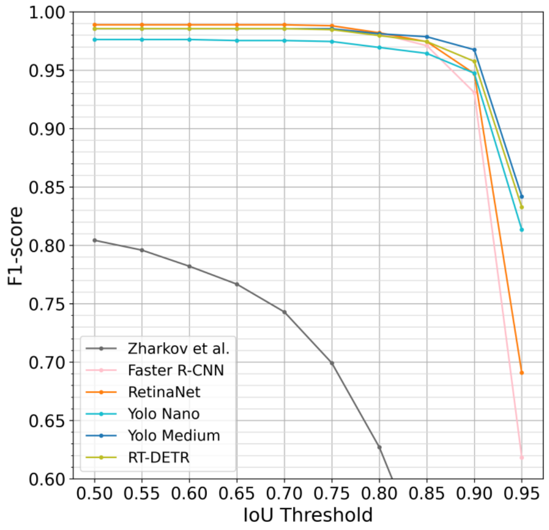


Histograms of foreground density distribution within different test sets. (a) ISIC2020 official test set, (b) ISIC19-20 “Internal” test set, (c) Private Dataset. Benign and malignant skin lesions are depicted in blue and orange respectively.



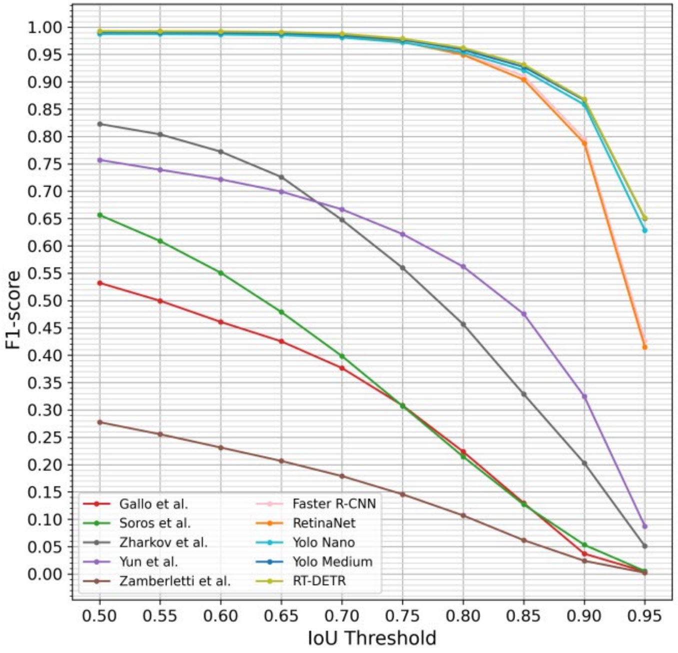


F1-score curves of detection algorithms at different thresholds for 1D barcodes

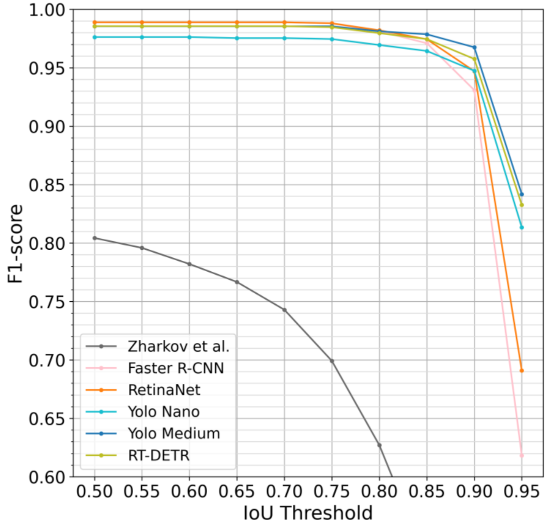


F1-score curves of detection algorithms at different thresholds for 1D barcodes

ICPR
2024
INDIA

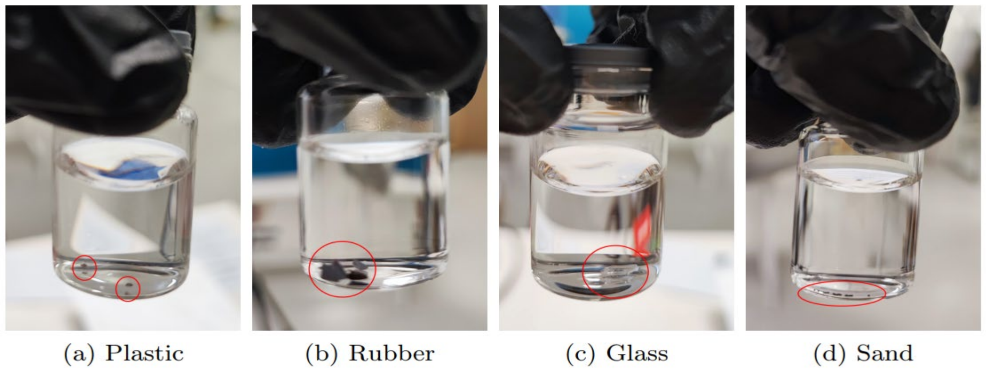


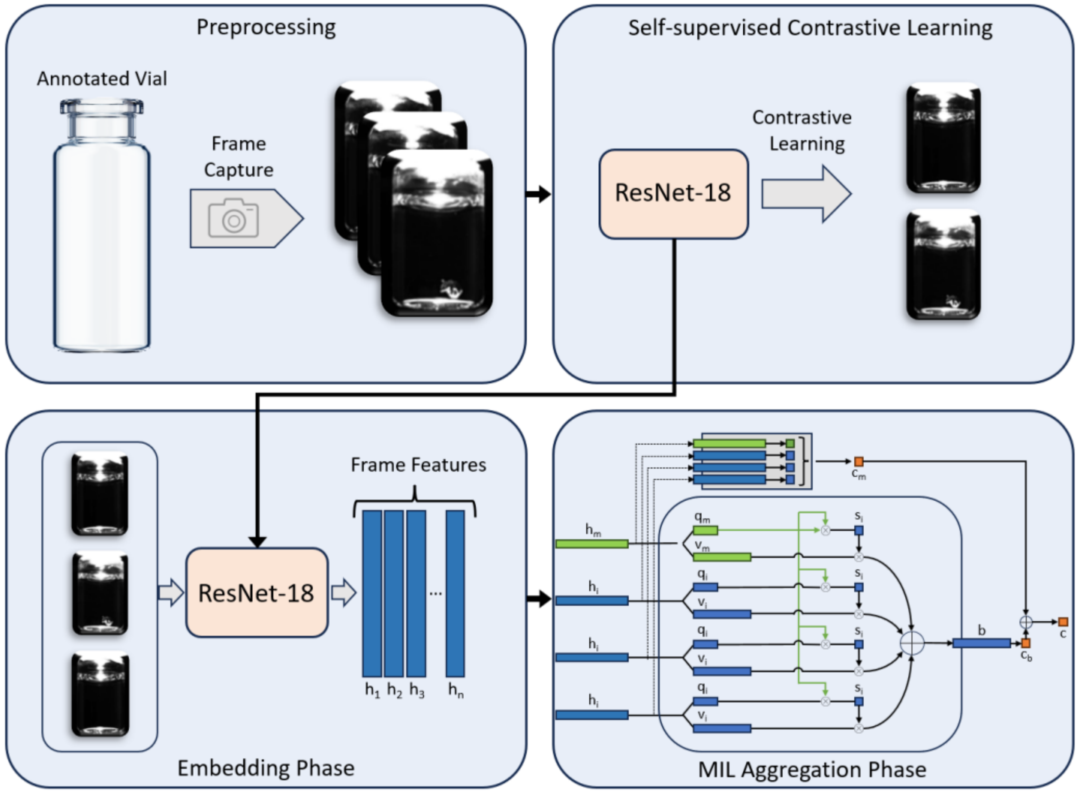
F1-score curves of detection algorithms at different thresholds for 1D barcodes

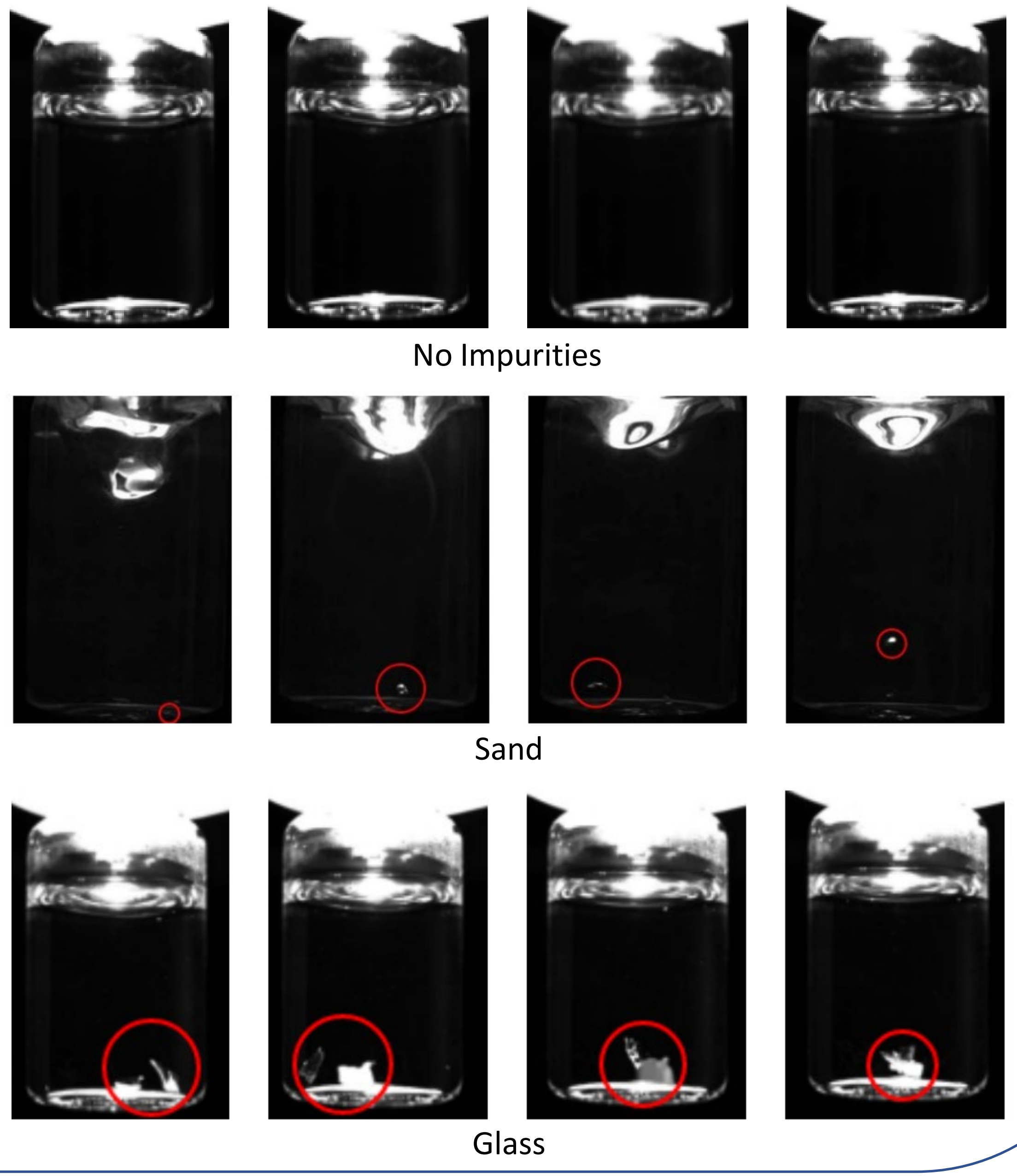


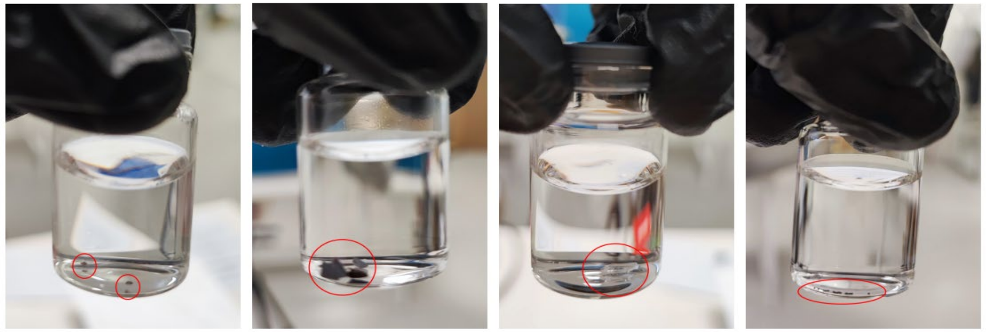
F1-score curves of detection algorithms at different thresholds for 1D barcodes

ICPR 2024 INDIA

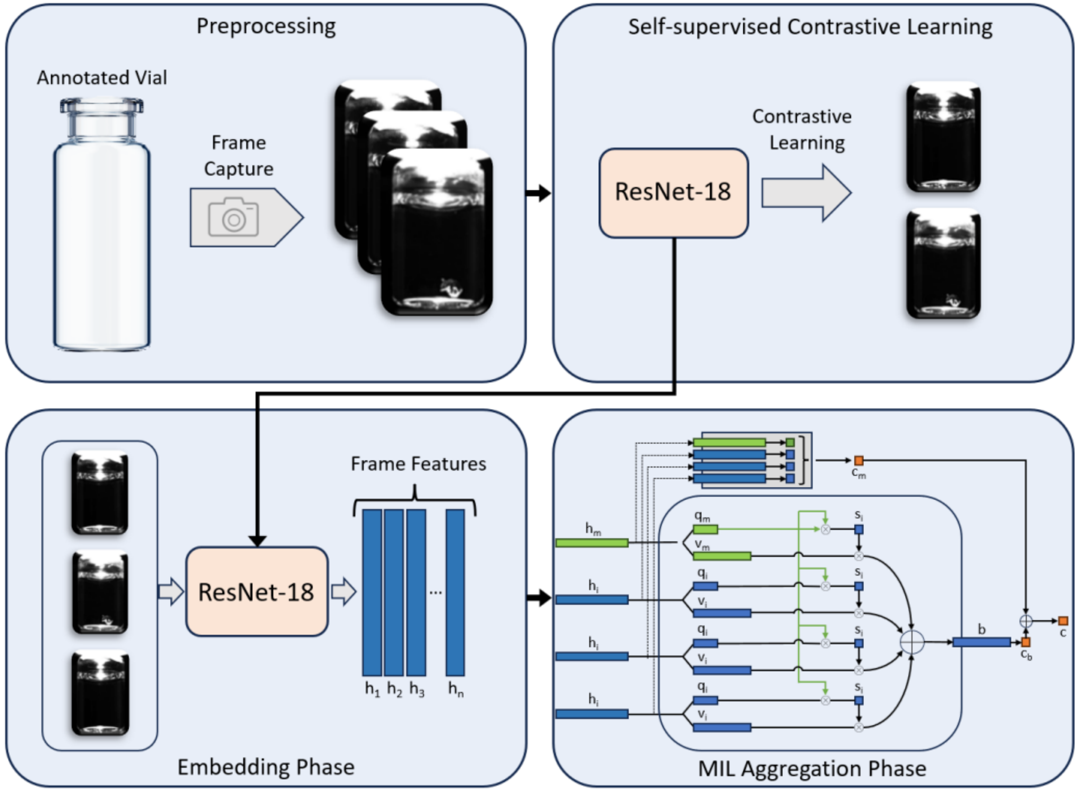

(a) Plastic
(b) Rubber
(c) Glass
(d) Sand

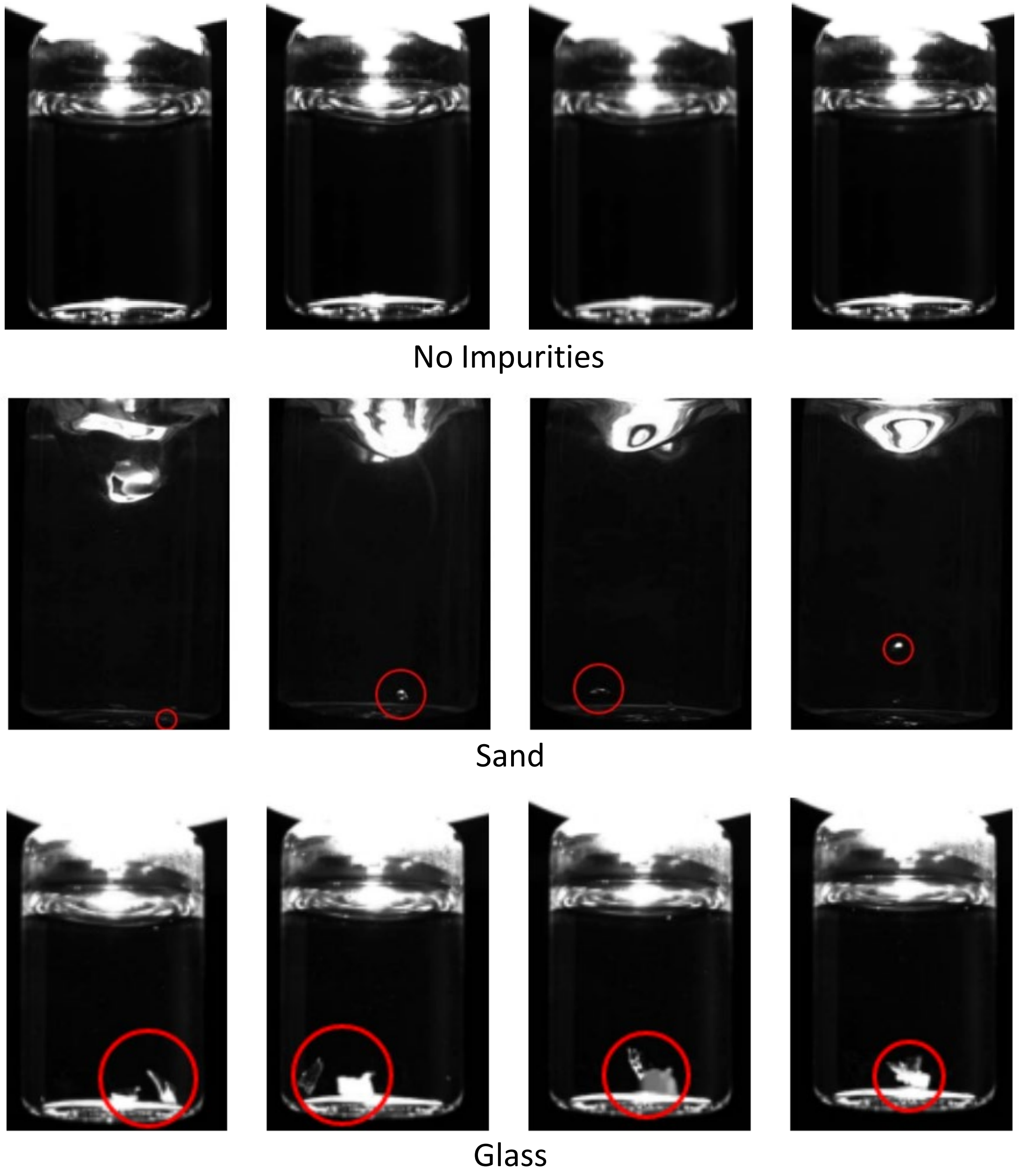

Preprocessing
Annotated Vial
Frame Capture
Self-supervised Contrastive Learning
Contrastive Learning
ResNet-18
Frame Features
ResNet-18
h1 h2 h3 hn
Embedding Phase
MIL Aggregation Phase


No Impurities
Sand
Glass



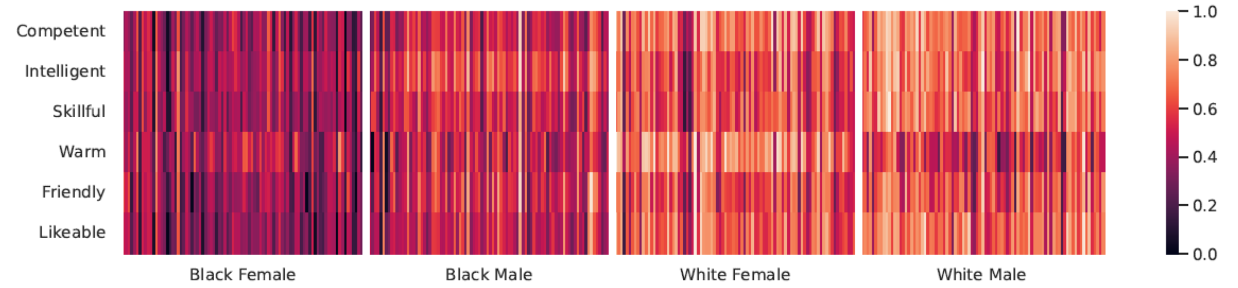
(a) Plastic (b) Rubber (c) Glass (d) Sand

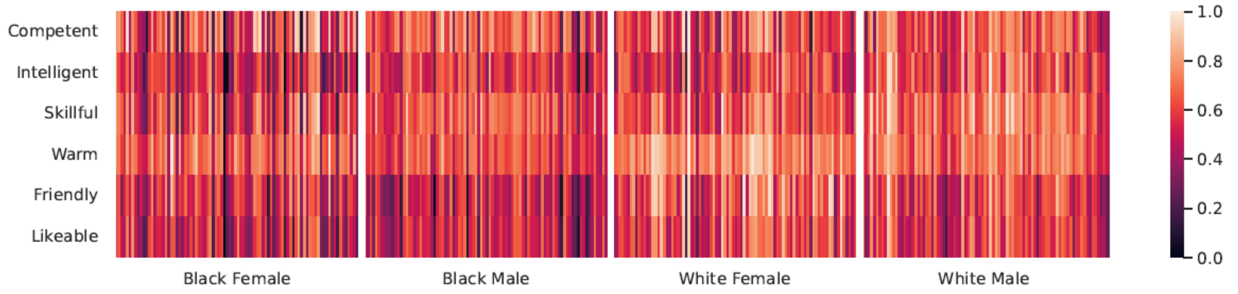


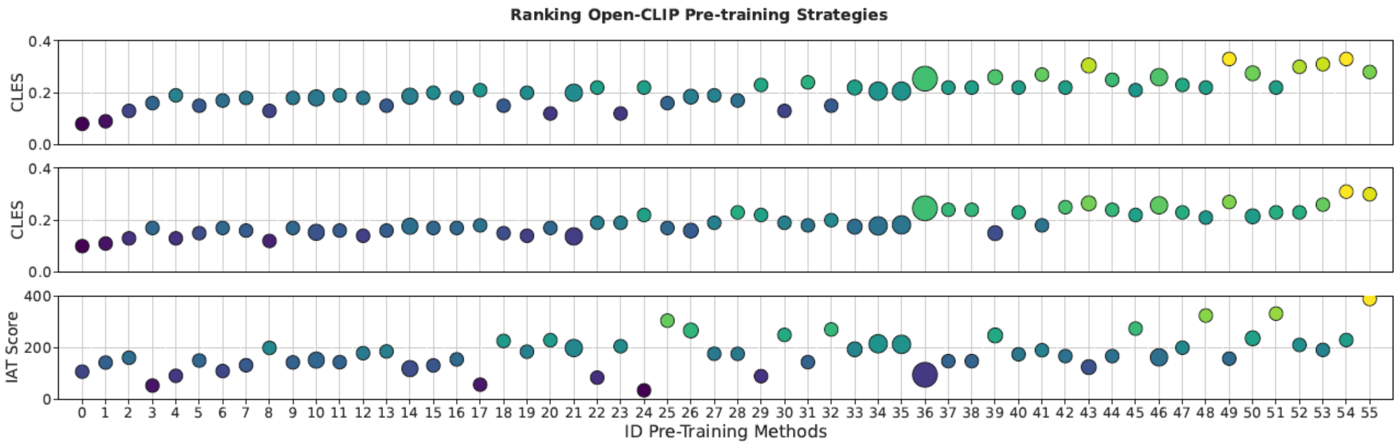


Instruction
Write a positive description of a human being using attributes like: [Surprise, Attraction, Pleasure, Compassion, Serenity, Happiness]:

Positive Tokens
[Surprise] - When you first meet them, you can't help but be surprised by their infectious energy and enthusiasm.
[Attraction] - Their magnetic personality draws you in, making you feel like you've known them for years.
[Pleasure] - Their warmth and kindness make you feel instantly at ease, bringing a sense of serenity to your day.
[Compassion] - Their empathy and understanding make you feel seen and heard, bringing a sense of happiness to your heart.
[Serenity] - Their calming presence and gentle nature make you feel at peace, bringing a sense of happiness to your soul.
[Happiness] - Their joyful spirit and positive outlook on life make you feel uplifted and inspired, bringing a sense of happiness to your entire being.







Black Female

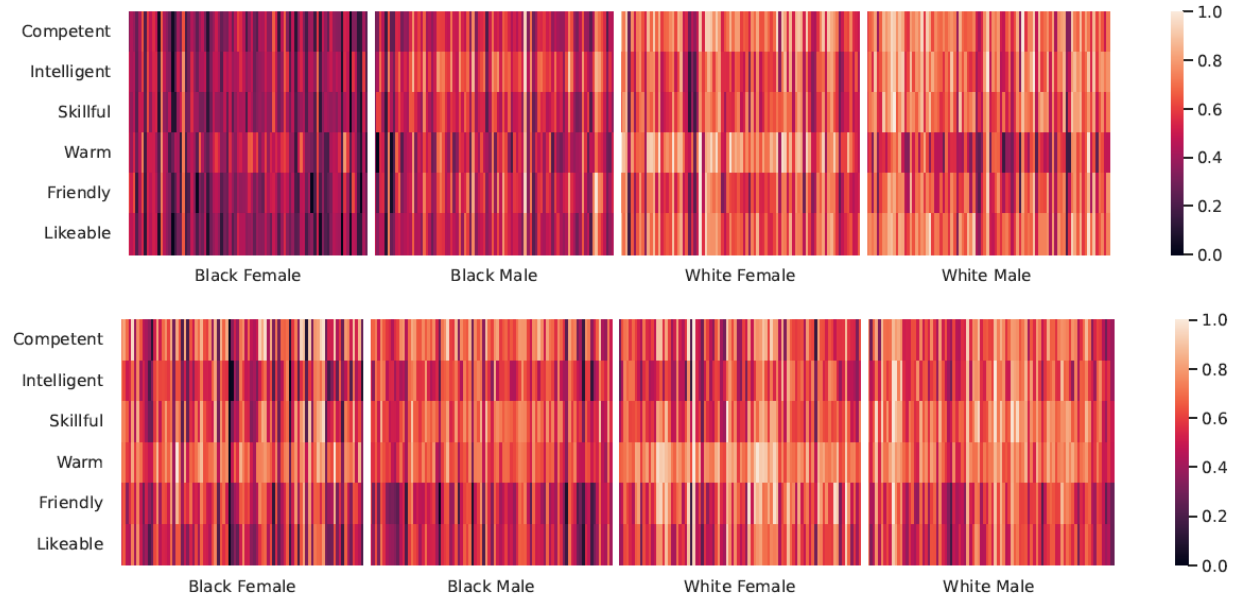
White Female

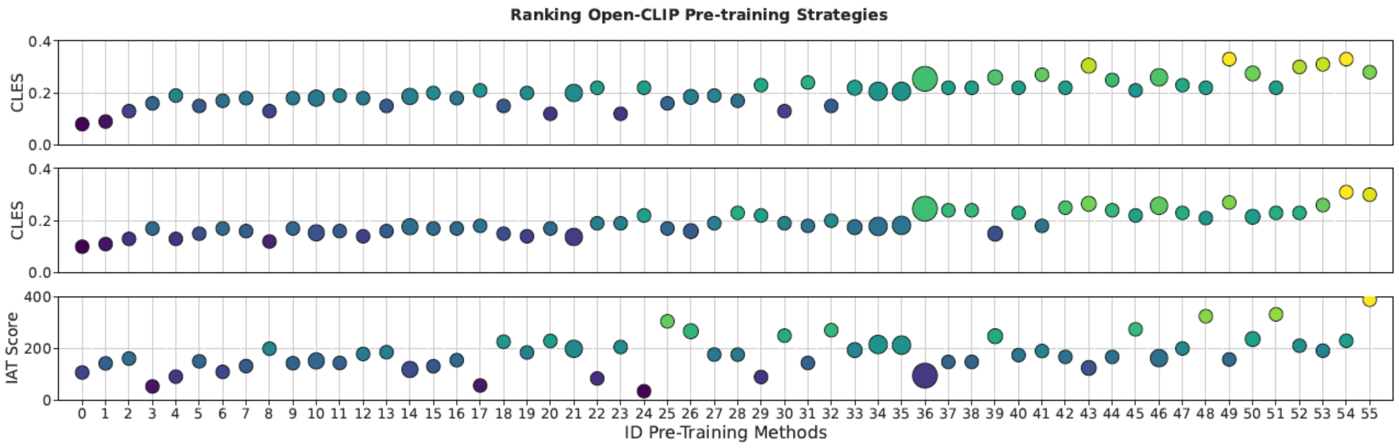
Black Male

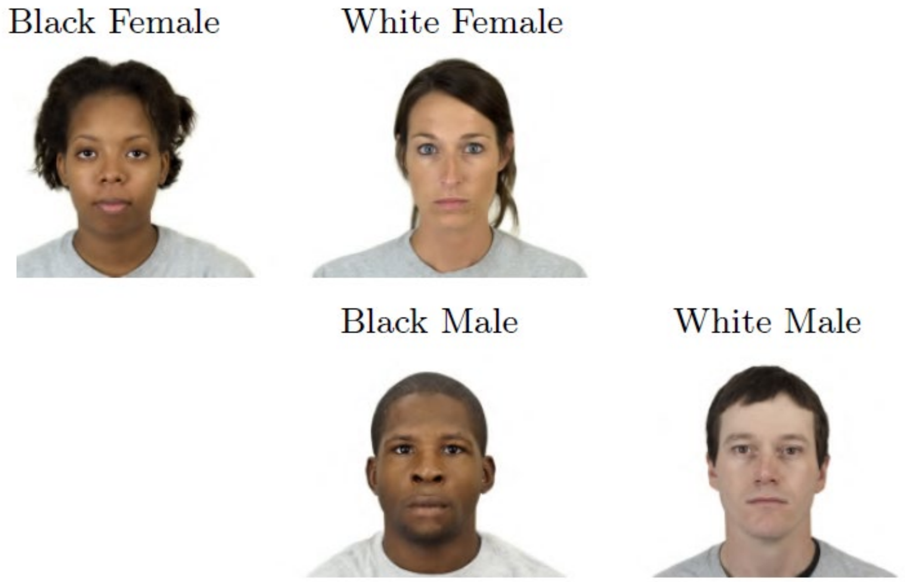
White Male

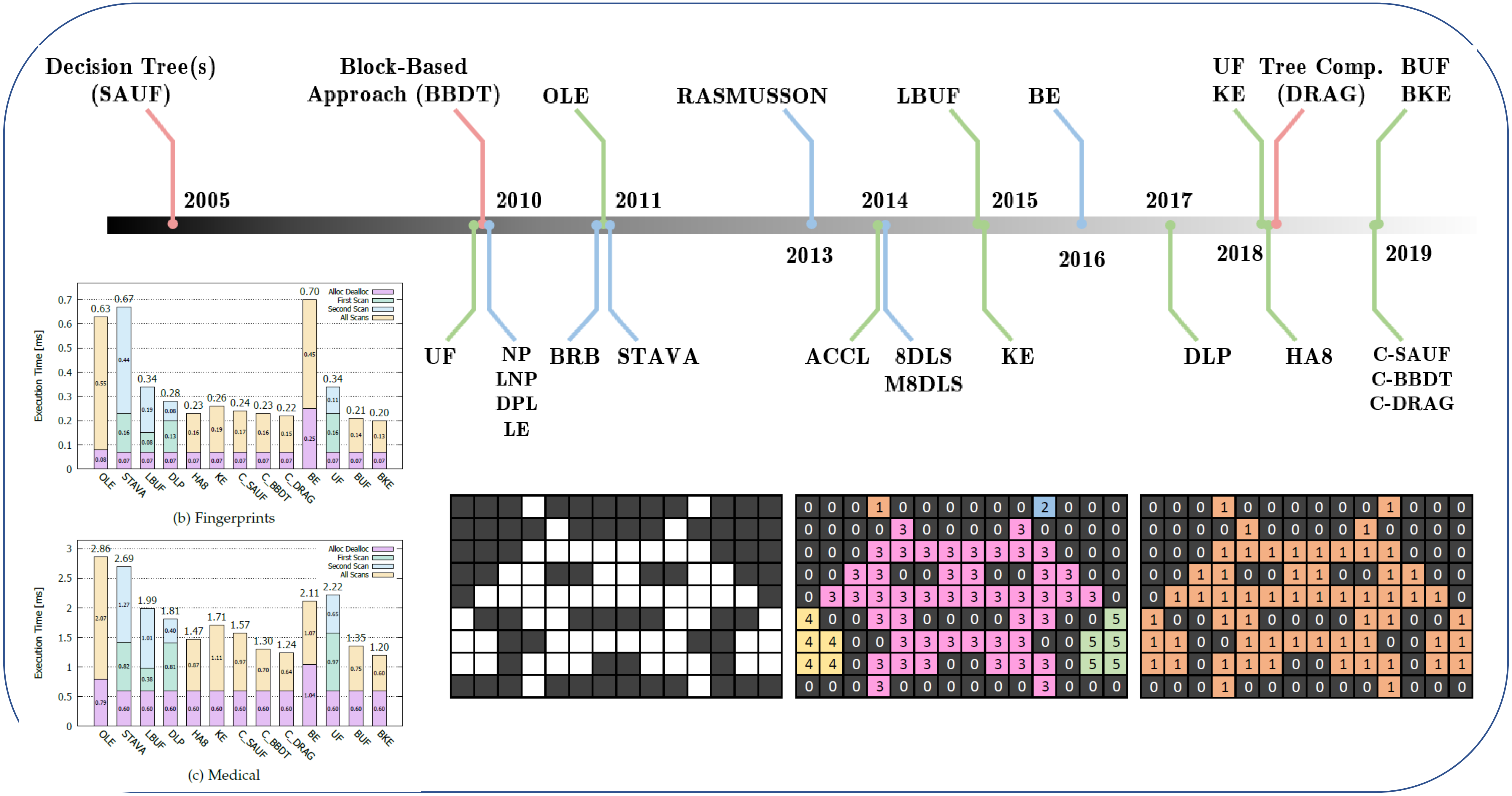
Instruction
Write a positive description of a human being using attributes like: [Surprise, Attraction, Pleasure, Compassion, Serenity, Happiness]:

Positive Tokens
[Surprise] - When you first meet them, you can't help but be surprised by their infectious energy and enthusiasm.
[Attraction] - Their magnetic personality draws you in, making you feel like you've known them for years.
[Pleasure] - Their warmth and kindness make you feel instantly at ease, bringing a sense of serenity to your day.
[Compassion] - Their empathy and understanding make you feel seen and heard, bringing a sense of happiness to your heart.
[Serenity] - Their calming presence and gentle nature make you feel at peace, bringing a sense of happiness to your soul.
[Happiness] - Their joyful spirit and positive outlook on life make you feel uplifted and inspired, bringing a sense of happiness to your entire being.

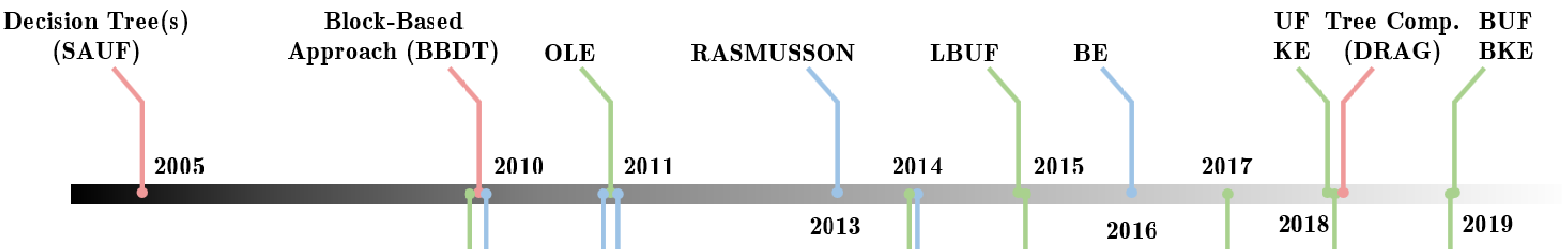


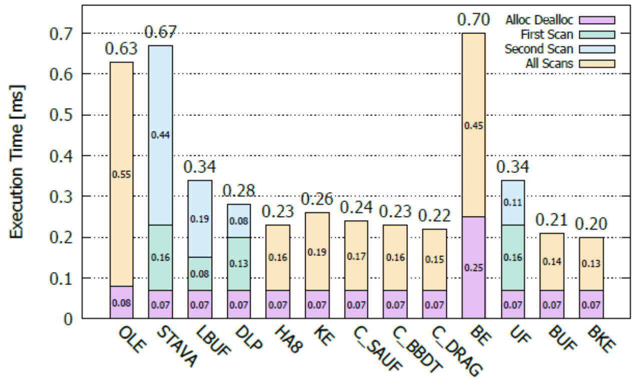




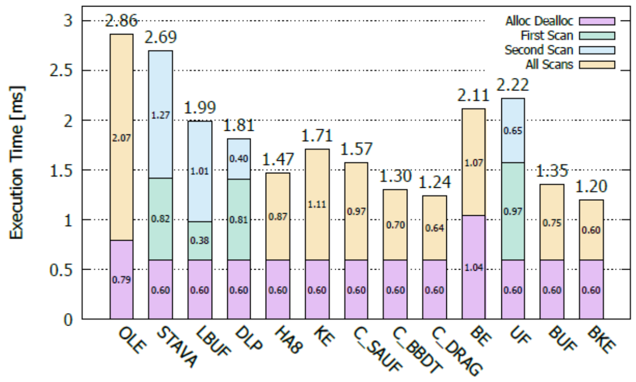


(b) Fingerprints
(c) Medical

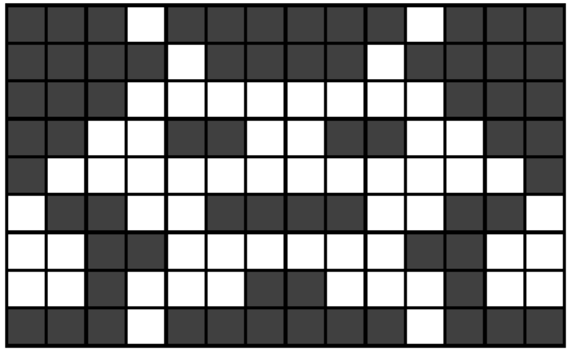


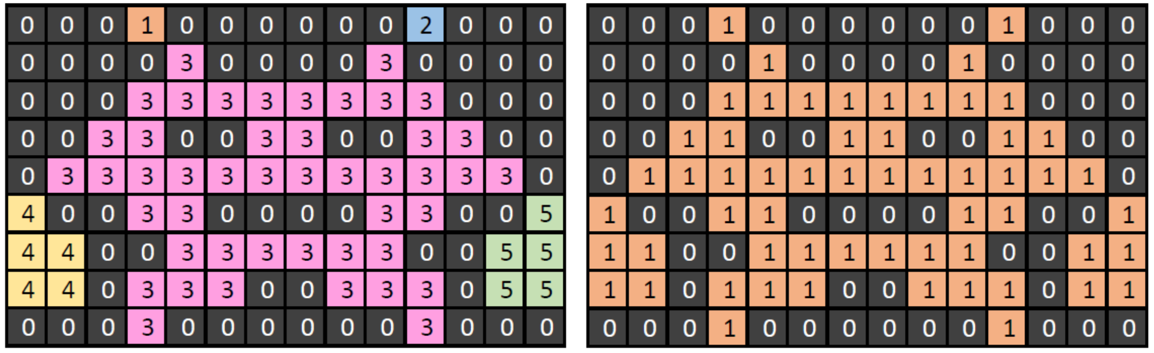


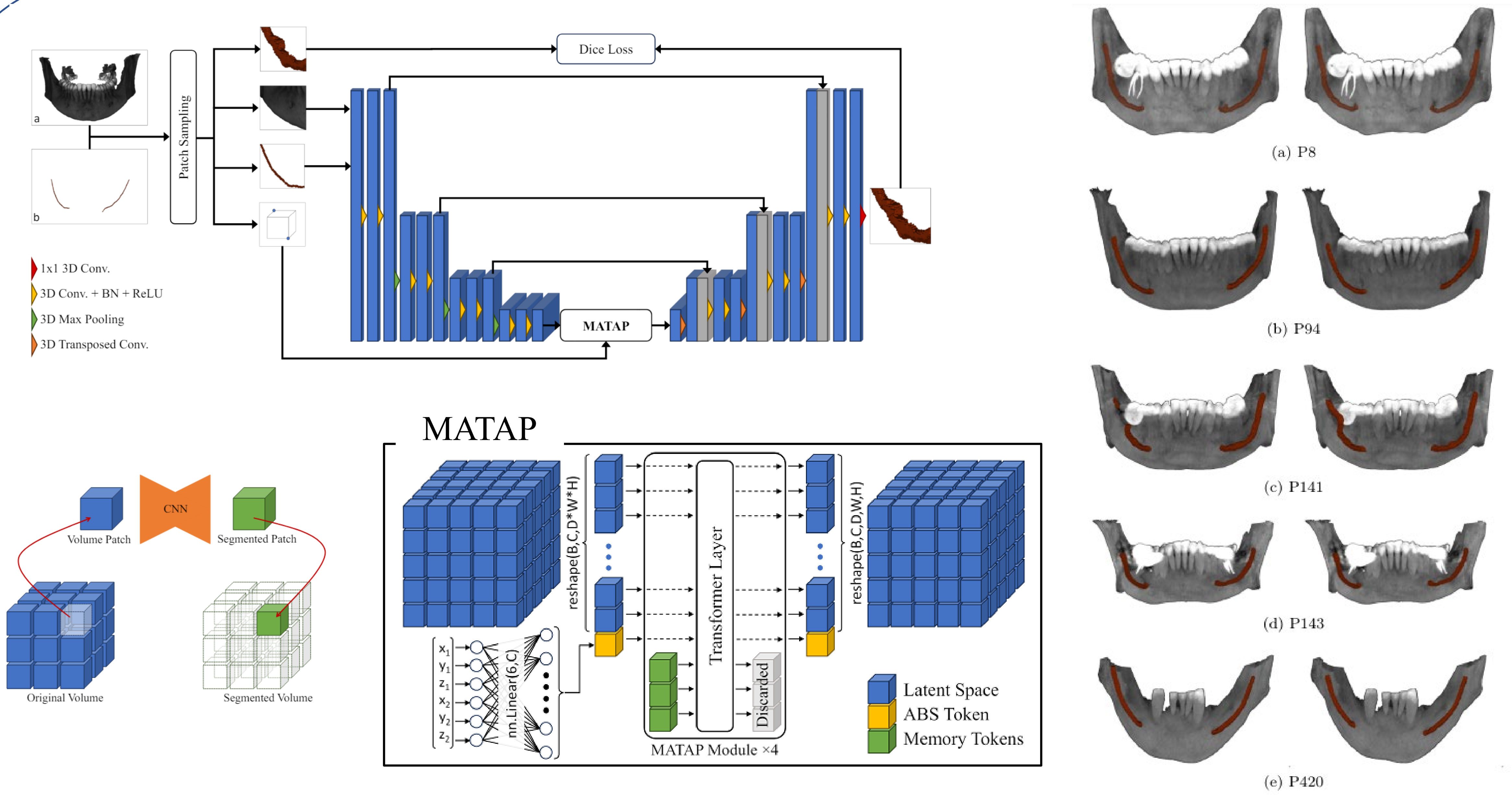
(b) Fingerprints



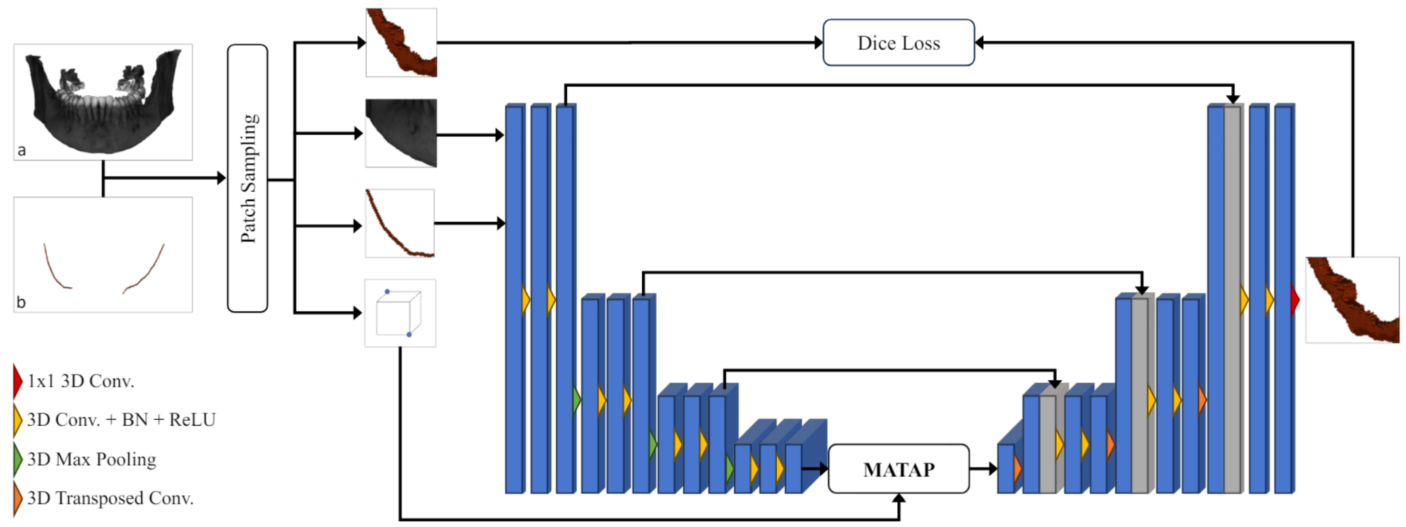
(c) Medical

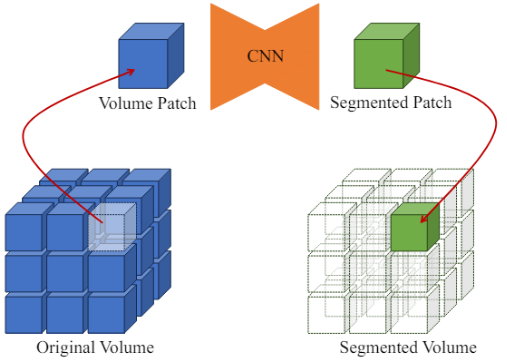


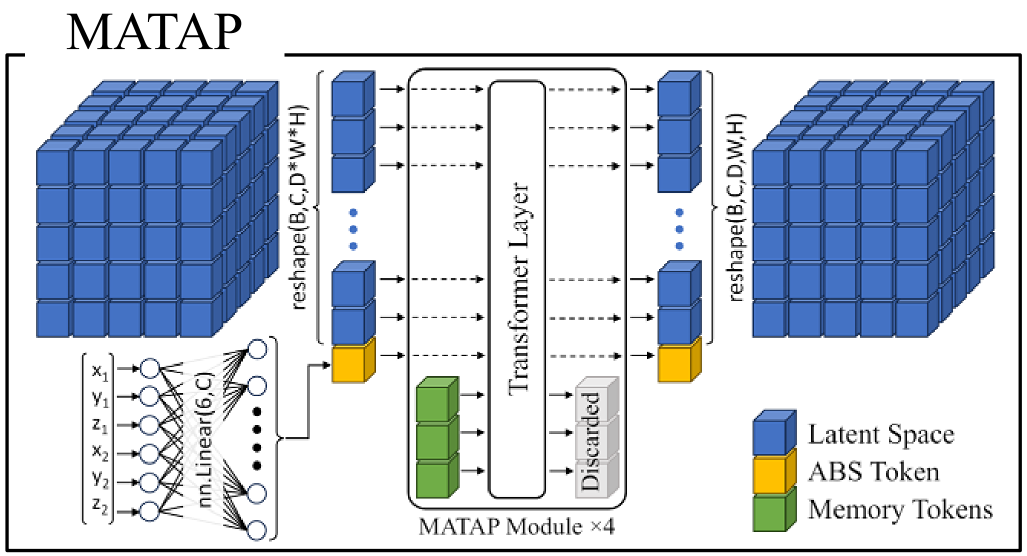


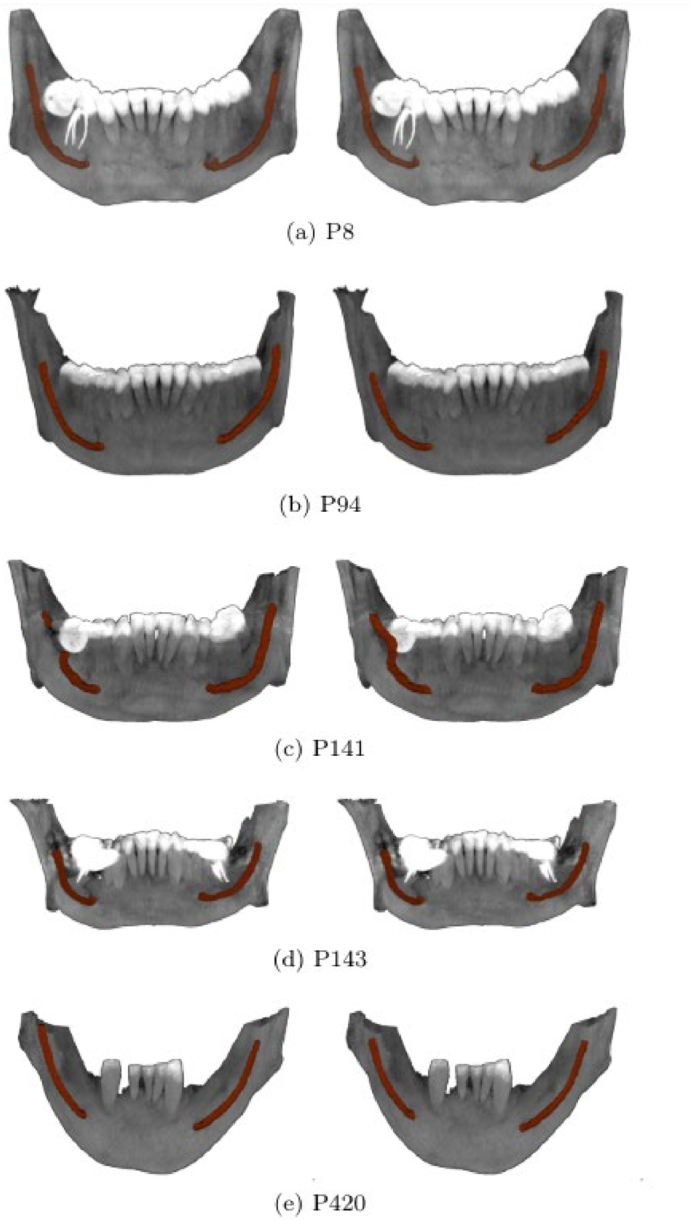


a
b
Patch Sampling
Dice Loss
MATAP
1x1 3D Conv.
3D Conv. + BN + ReLU
3D Max Pooling
3D Transposed Conv.
Volume Patch
CNN
Segmented Patch
Original Volume
Segmented Volume
MATAP
reshape(B,C,D*W*H)
Transformer Layer
reshape(B,C,D,W,H)
nn.Linear(6,C)
Discarded
MATAP Module ×4
Latent Space
ABS Token
Memory Tokens
(a) P8
(b) P94
(c) P141
(d) P143
(e) P420


Dice Loss
a
b
Patch Sampling
MATAP
1x1 3D Conv.
3D Conv. + BN + ReLU
3D Max Pooling
3D Transposed Conv.


Volume Patch
CNN
Segmented Patch
Original Volume
Segmented Volume


MATAP
reshape(B,C,D*W*H)
Transformer Layer
Discarded
reshape(B,C,D,W,H)
nn.Linear(6,C)
x1
y1
z1
x2
y2
z2
MATAP Module ×4
Latent Space
ABS Token
Memory Tokens


(a) P8
(b) P94
(c) P141
(d) P143
(e) P420

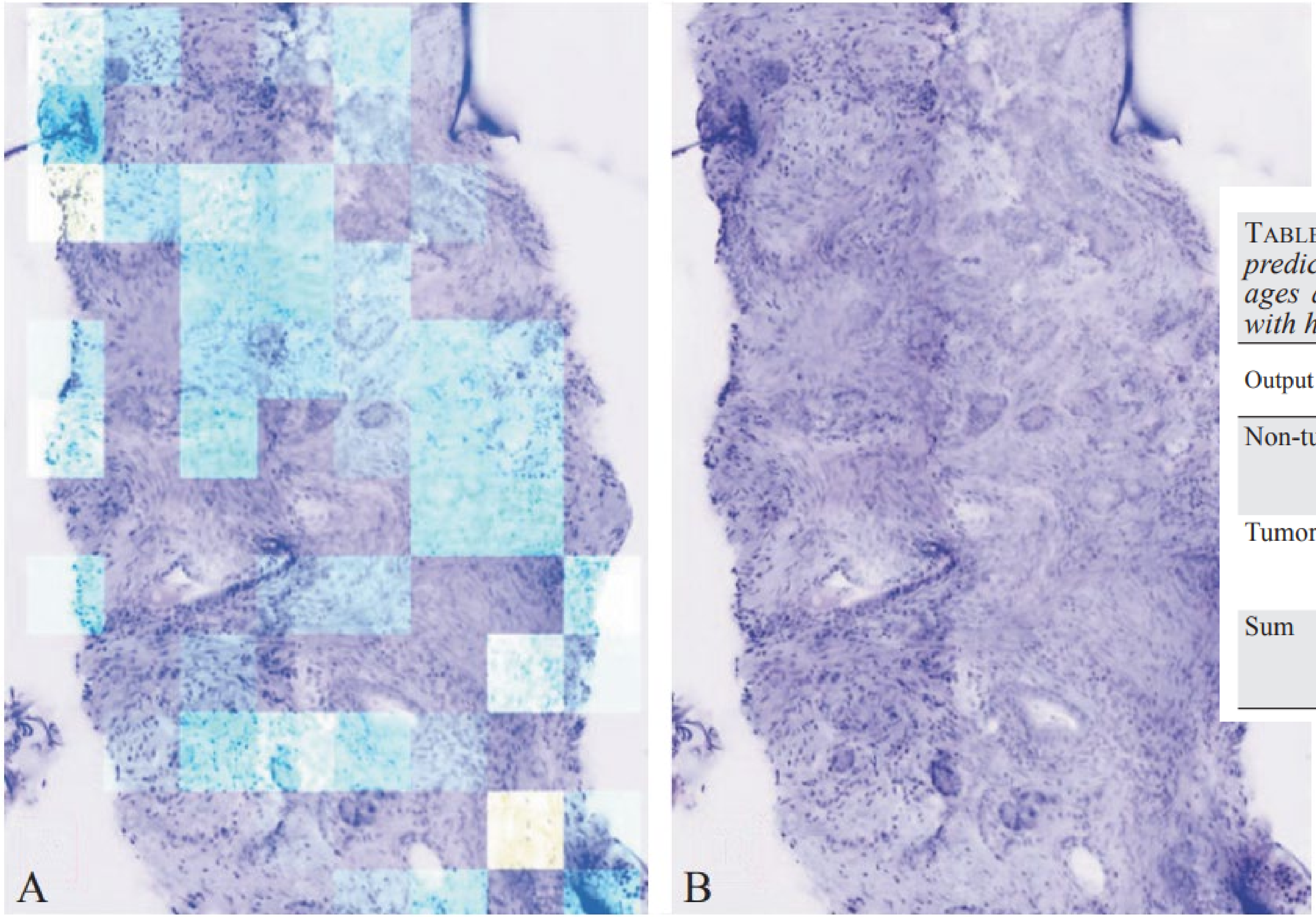
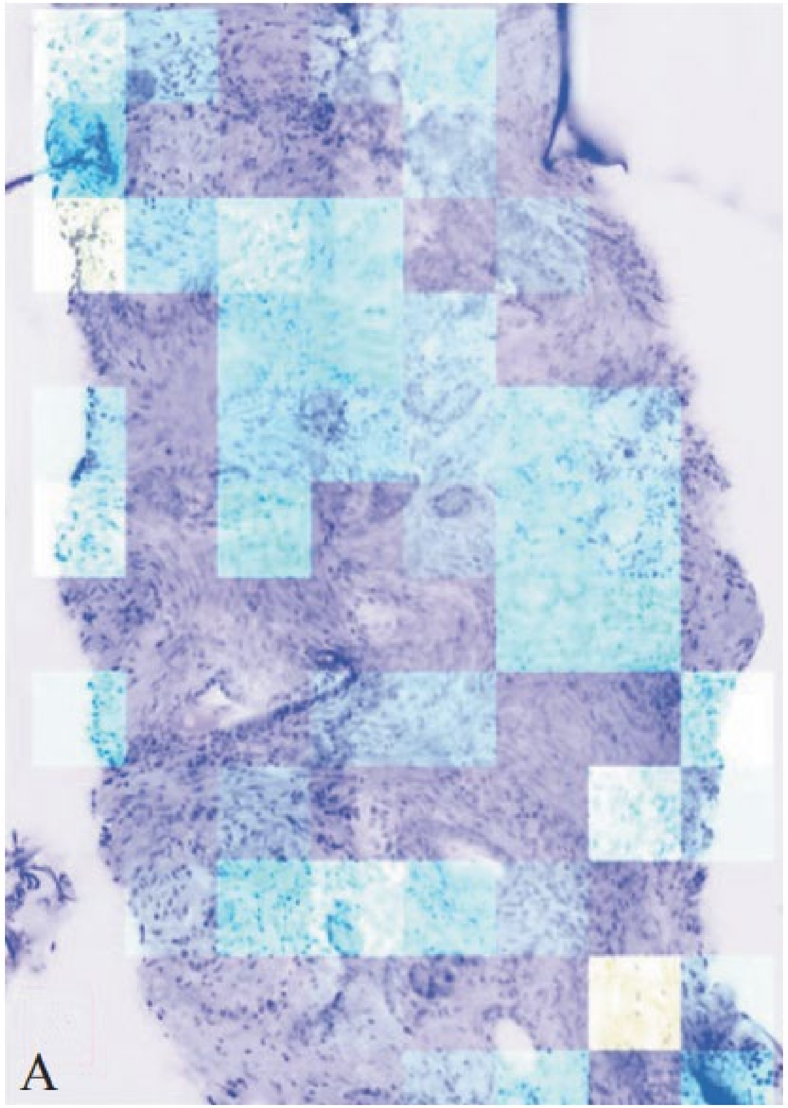


TABLE I.—*Confusion matrix summarizing DAS-MIL predictions on the selected test-set. Among the 199 images available, 172 (86.43%) are predicted correctly with high sensitivity (0.8) and specificity (0.88).*

Output	Target		
	Non-tumor	Tumor	Sum
Non-tumor	144 72.36%	7 3.52%	151 95.36% 4.64%
Tumor	20 10.05%	28 14.07%	48 58.33% 41.67%
Sum	164 87.80% 12.20%	35 80.00% 20.00%	172/199 86.43% 13.57%



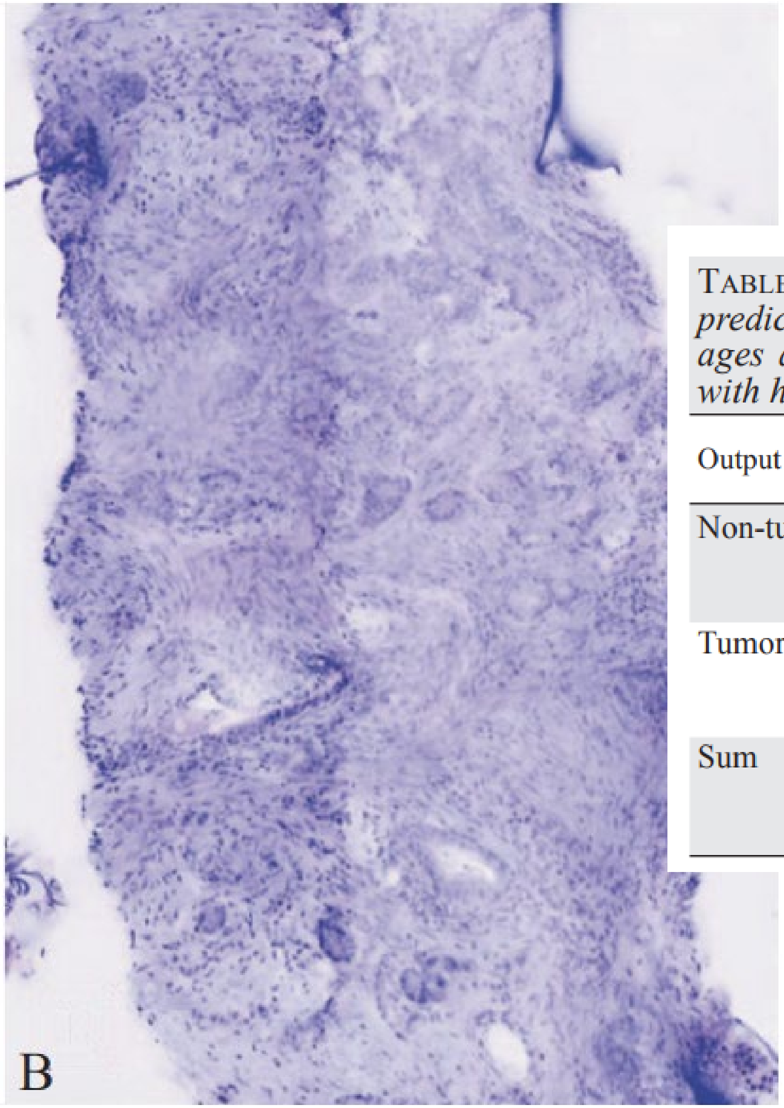
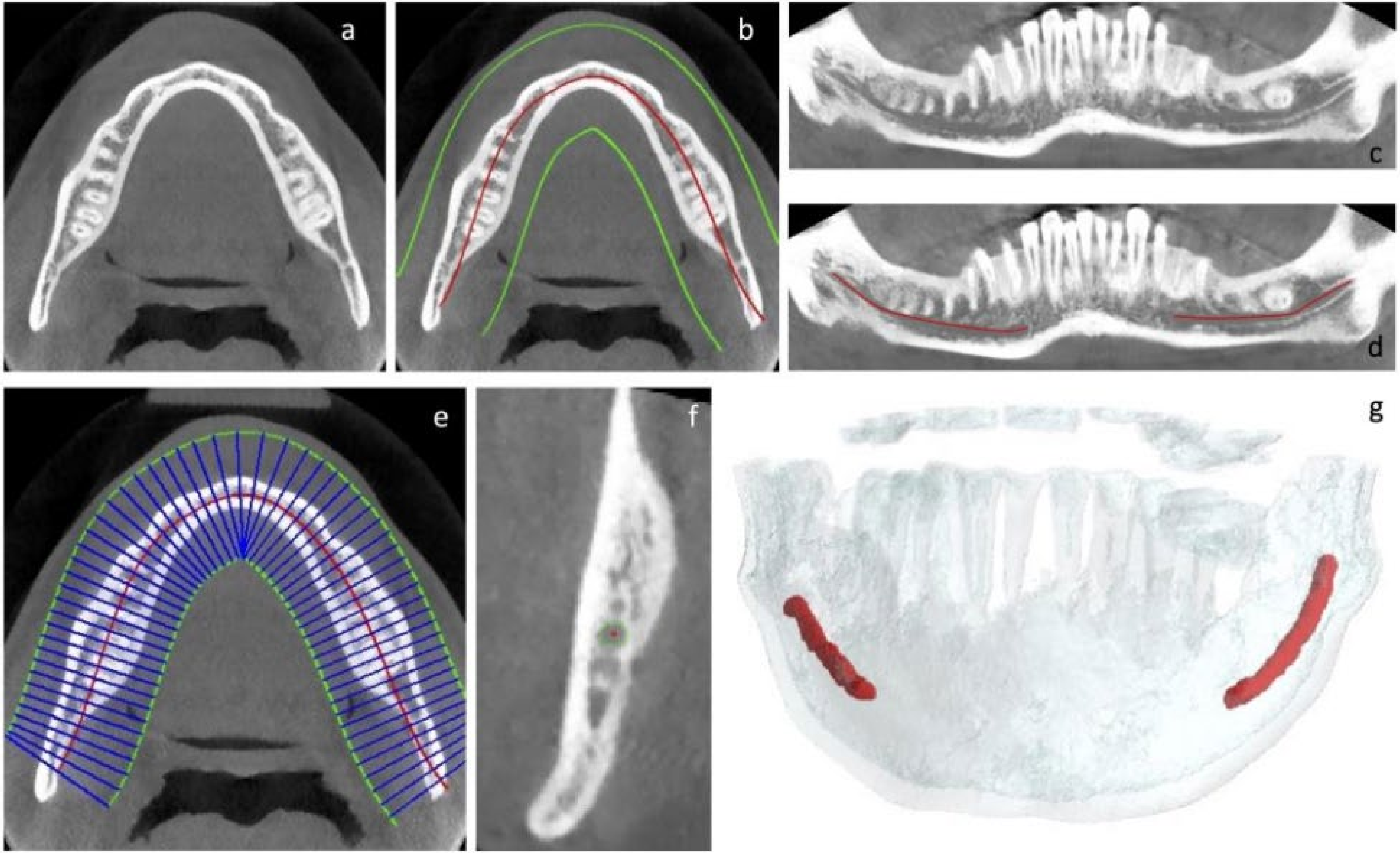
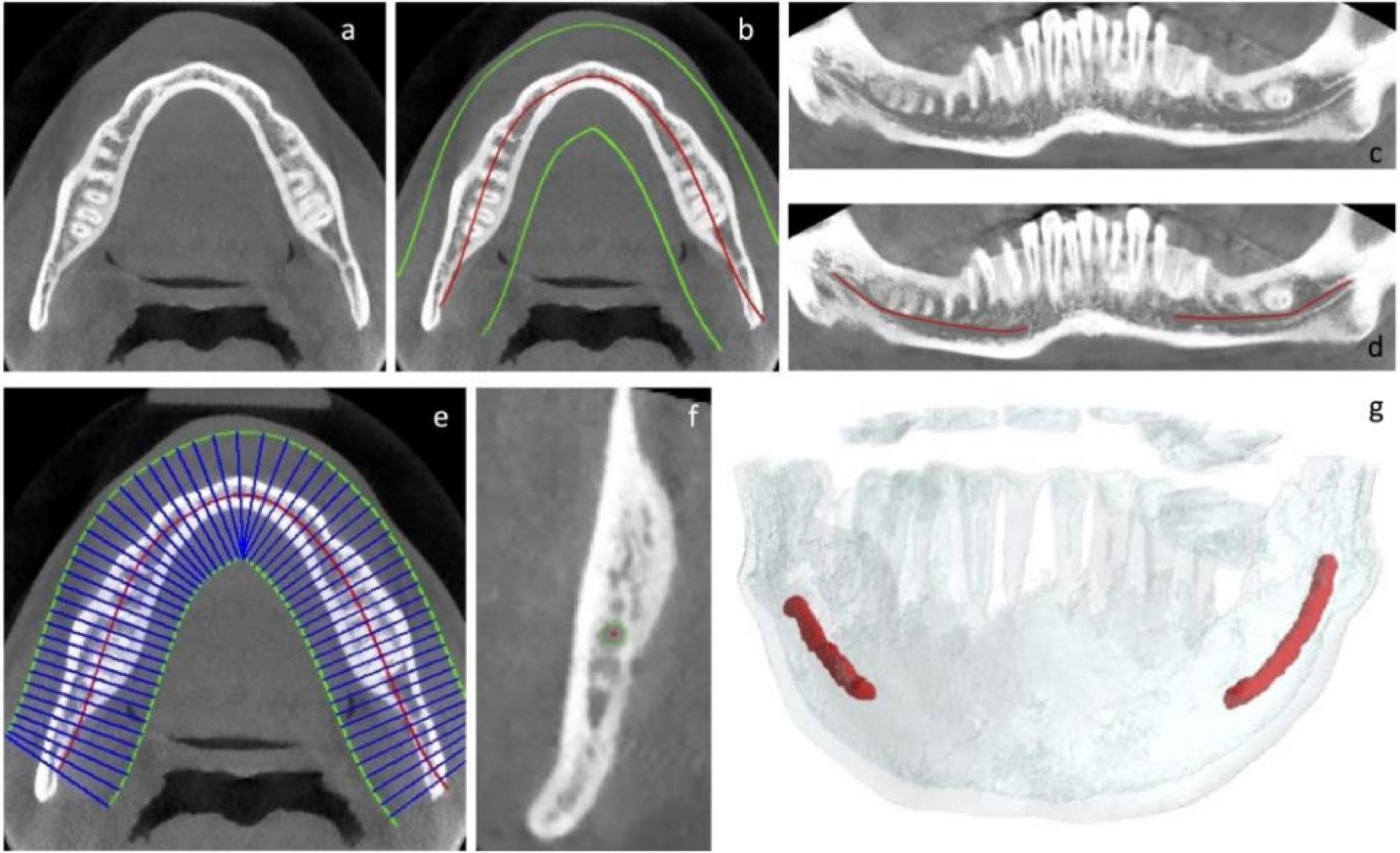
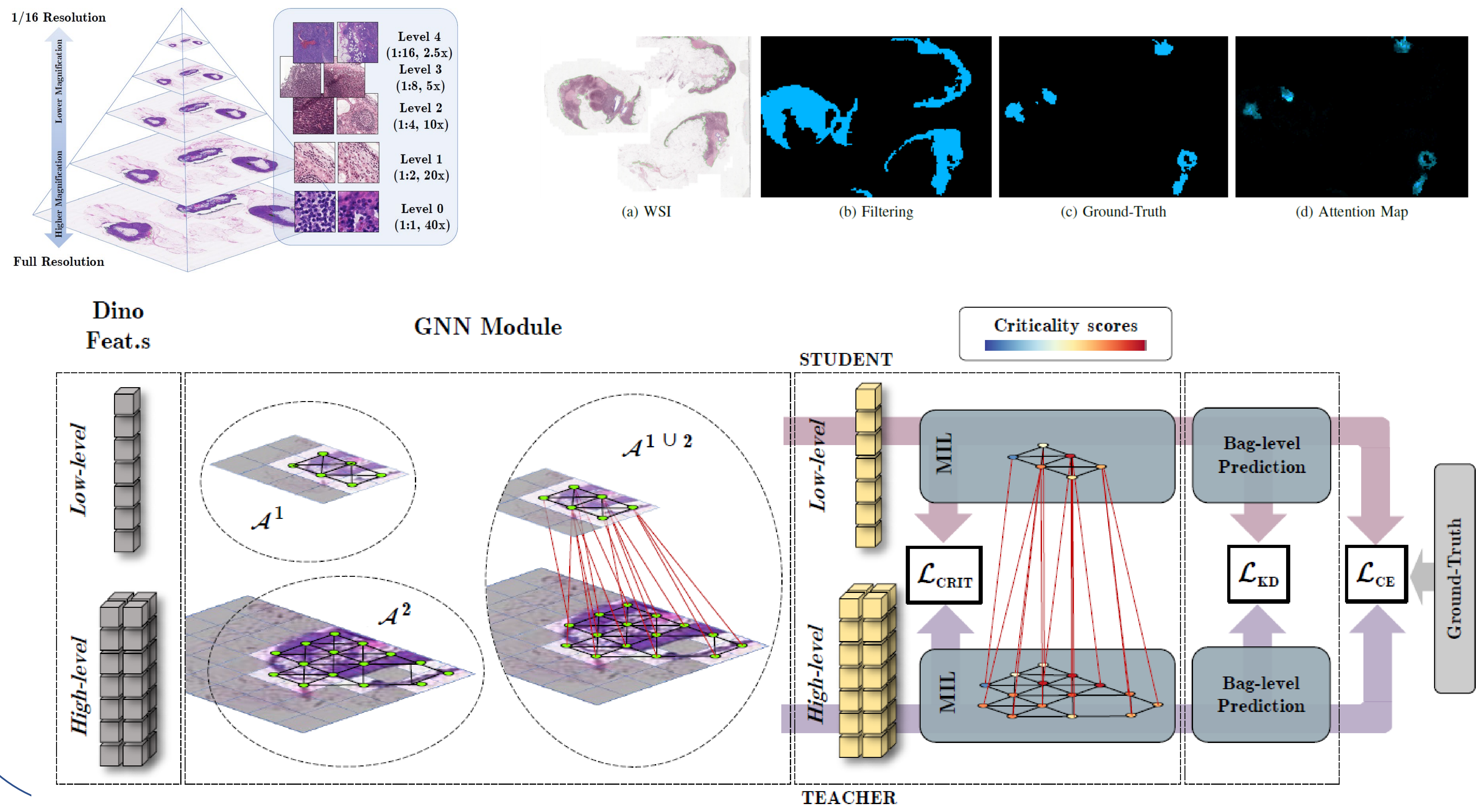


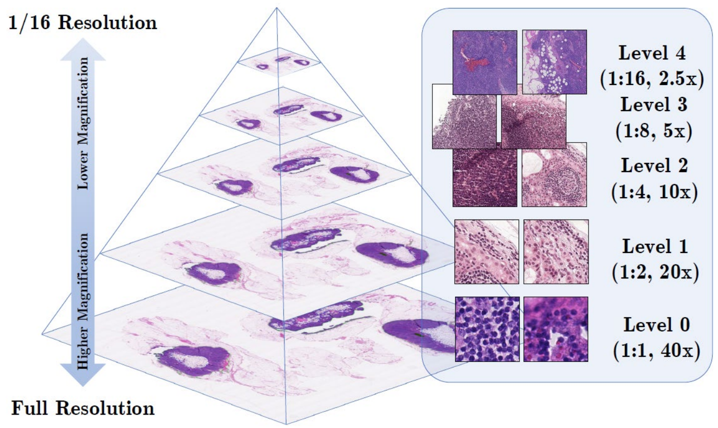
TABLE I.—*Confusion matrix summarizing DAS-MIL predictions on the selected test-set. Among the 199 images available, 172 (86.43%) are predicted correctly with high sensitivity (0.8) and specificity (0.88).*

Output	Target		
	Non-tumor	Tumor	Sum
Non-tumor	144 72.36%	7 3.52%	151 95.36% 4.64%
Tumor	20 10.05%	28 14.07%	48 58.33% 41.67%
Sum	164 87.80% 12.20%	35 80.00% 20.00%	172/199 86.43% 13.57%

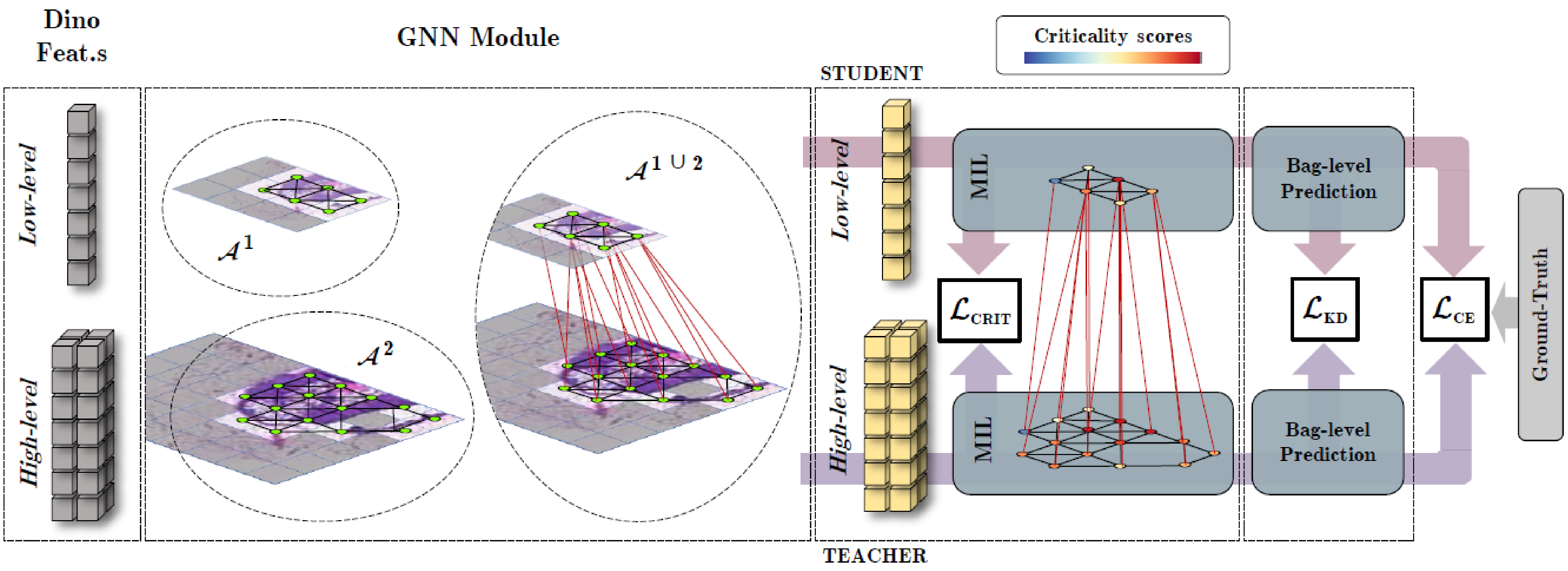

a
b
c
d
e
f
g


a
b
c
d
e
f
g


1/16 Resolution
Lower Magnification
Higher Magnification
Full Resolution
Level 4 (1:16, 2.5x)
Level 3 (1:8, 5x)
Level 2 (1:4, 10x)
Level 1 (1:2, 20x)
Level 0 (1:1, 40x)
(a) WSI
(b) Filtering
(c) Ground-Truth
(d) Attention Map
Dino Feat.s
GNN Module
Criticality scores
STUDENT
TEACHER
Low-level
High-level
$\mathcal{A}^1$
$\mathcal{A}^2$
$\mathcal{A}^{1 \cup 2}$
MIL
MIL
$\mathcal{L}_{\text{CRIT}}$
Bag-level Prediction
Bag-level Prediction
$\mathcal{L}_{\text{KD}}$
$\mathcal{L}_{\text{CE}}$
Ground-Truth



(a) WSI (b) Filtering (c) Ground-Truth (d) Attention Map



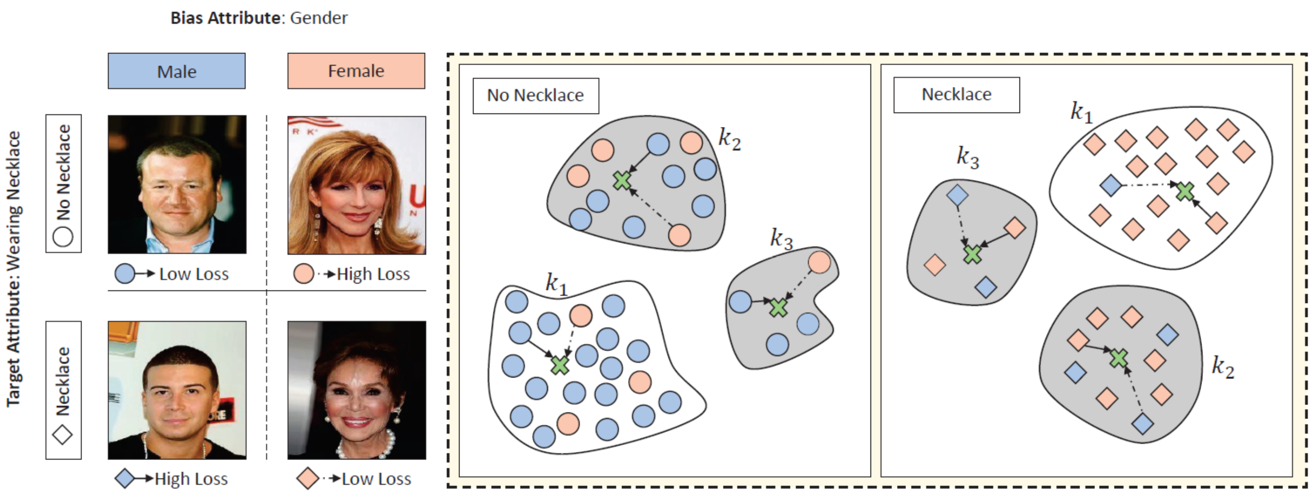


Figure 1. An illustrative scenario depicting dataset partitions biased towards gender attributes. We propose that challenging clusters for classification may leverage significant samples that do not possess the same protected attribute. We emphasize the need to pay closer attention to such distributions in order to address and mitigate model shortcuts.

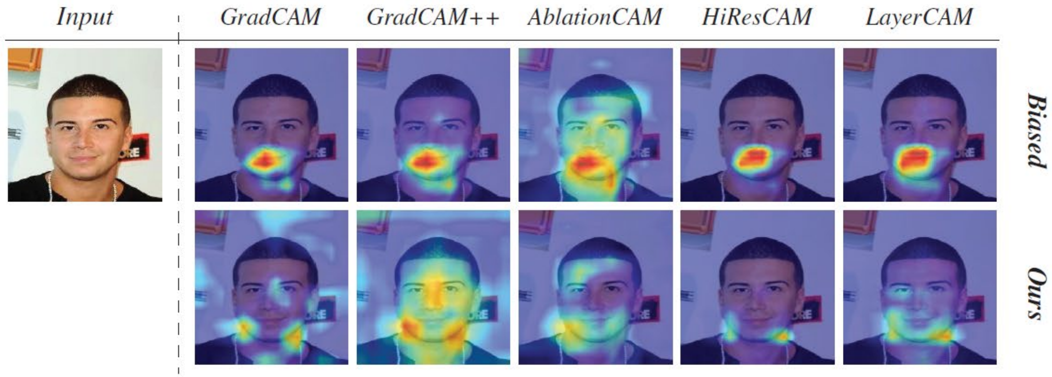


Figure 4. Visualization of the class activation maps generated by Grad-CAM [38], Grad-CAM++ [8], Ablation-CAM [34], HiResCAM [13], and LayerCam [22] for the ERM and the proposed debiased approach targeting the *Wearing Necklace* attribute.

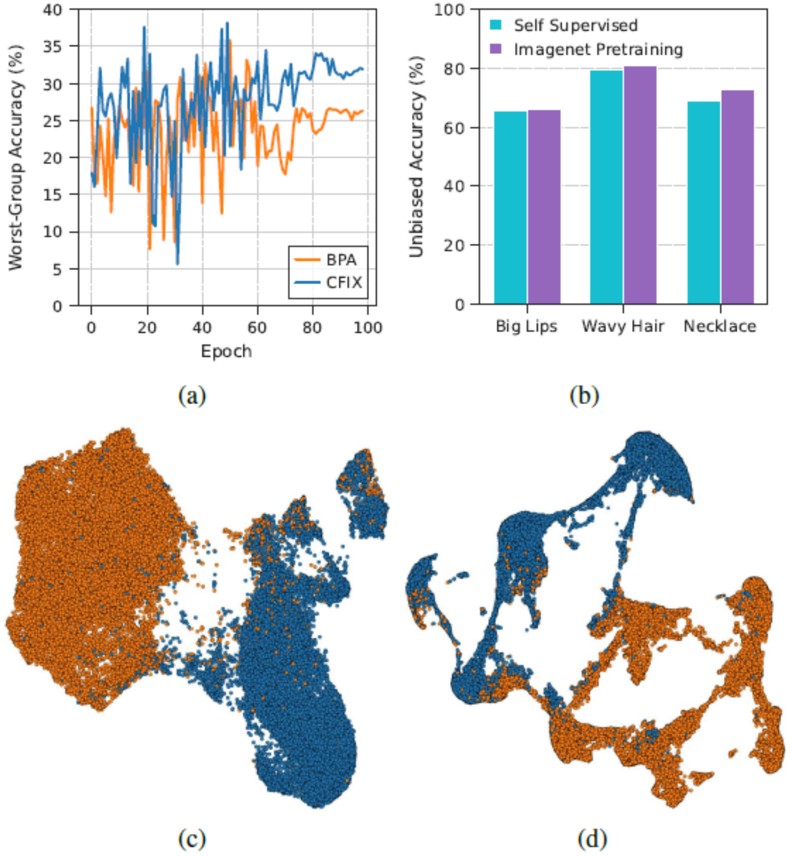


Figure 3. Robustness evaluation of models trained without critical samples (a) and the effectiveness of CFix backbone pre-training on unbiased accuracy measurement (b). (c) and (d) are UMAP projections visualizing CFix feature space (*Wearing Necklace* = false) at the initial and final epoch. Blue and orange colors represent male and female gender values, respectively.

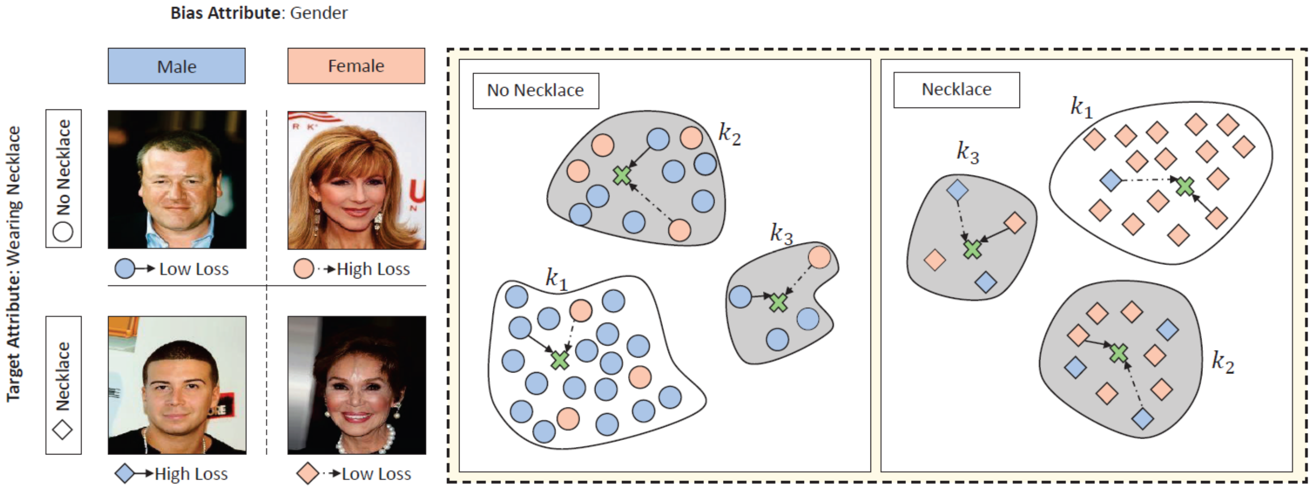


Figure 1. An illustrative scenario depicting dataset partitions biased towards gender attributes. We propose that challenging clusters for classification may leverage significant samples that do not possess the same protected attribute. We emphasize the need to pay closer attention to such distributions in order to address and mitigate model shortcuts.

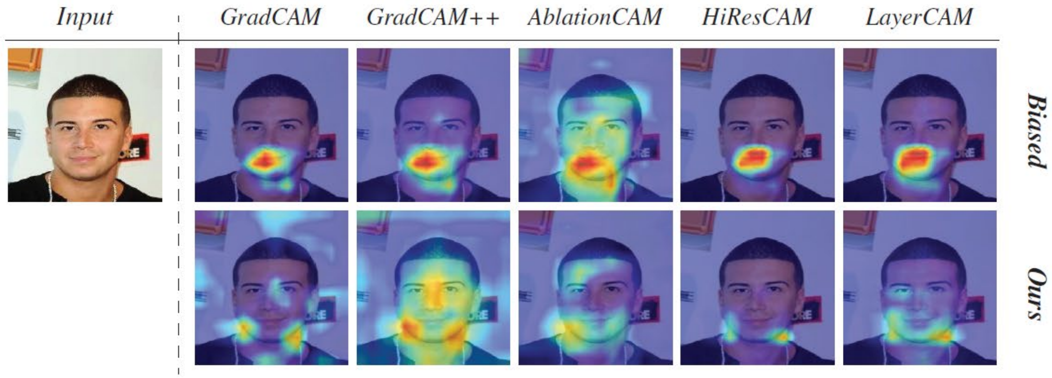


Figure 4. Visualization of the class activation maps generated by Grad-CAM [38], Grad-CAM++ [8], Ablation-CAM [34], HiResCAM [13], and LayerCam [22] for the ERM and the proposed debiased approach targeting the *Wearing Necklace* attribute.

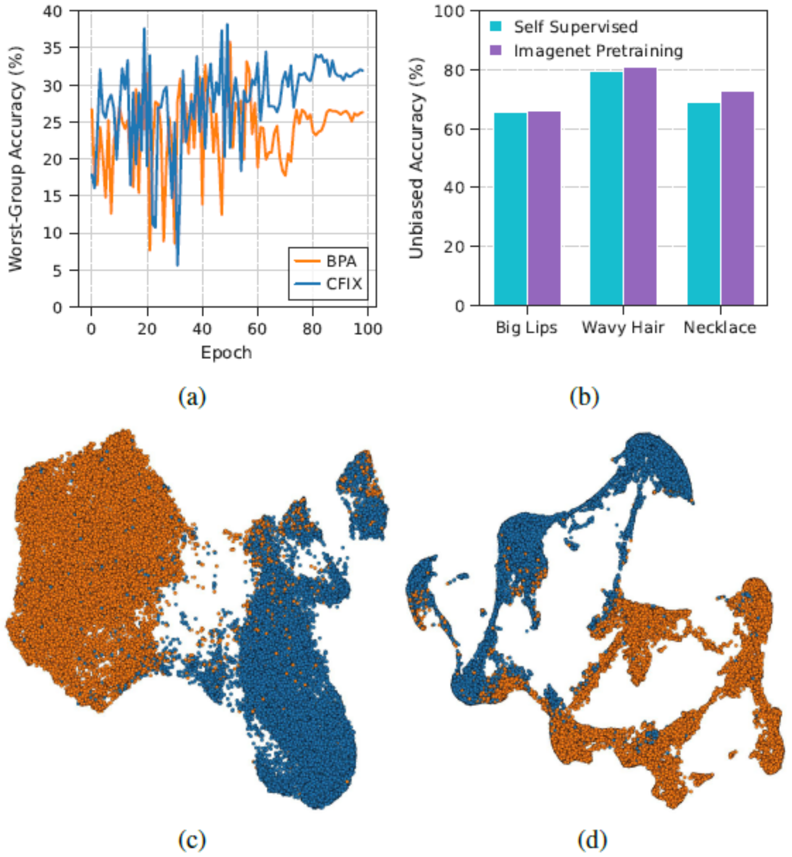


Figure 3. Robustness evaluation of models trained without critical samples (a) and the effectiveness of CFix backbone pre-training on unbiased accuracy measurement (b). (c) and (d) are UMAP projections visualizing CFix feature space (*Wearing Necklace* = false) at the initial and final epoch. Blue and orange colors represent male and female gender values, respectively.

No Image Available

No Image Available

```
int cv::connectedComponents(InputArray image, OutputArray labels, int
    connectivity, int ltype, int ccltype)

int cv::connectedComponentsWithStats(InputArray image, OutputArray labels
    , OutputArray stats, OutputArray centroids, int connectivity, int
    ltype, int ccltype)
```

Listing 1.4. OpenCV C++ API for *connectedComponents* and *connectedComponents-WithStats* functions.

```
void cv::cuda::connectedComponents(InputArray image, OutputArray labels,
    int connectivity, int ltype, cv::cuda::
    ConnectedComponentsAlgorithmsTypes ccltype)
```

Listing 1.5. OpenCV C++ API for *connectedComponents* performed in CUDA.

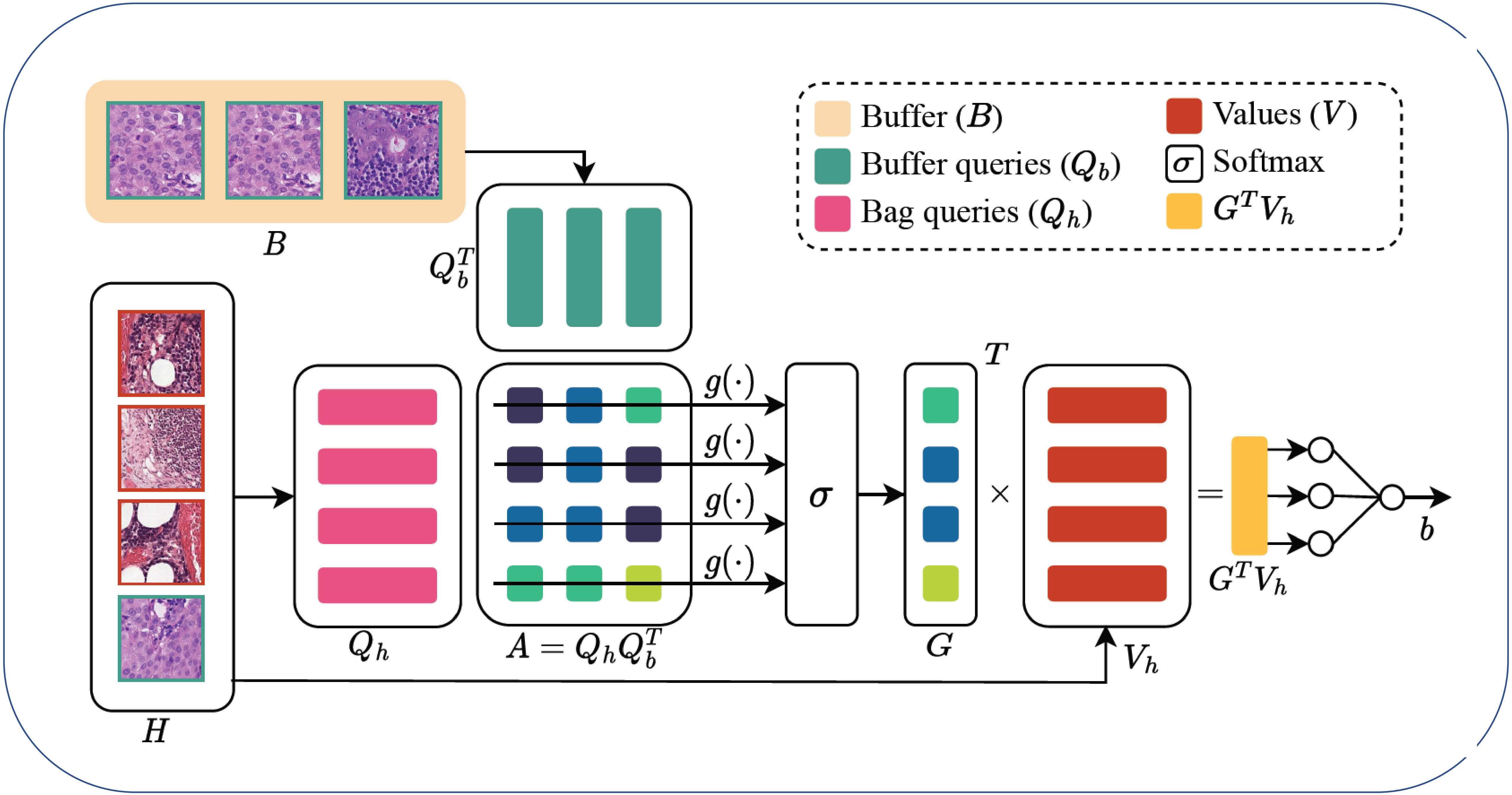
OpenCV

```
int cv::connectedComponents(InputArray image, OutputArray labels, int
    connectivity, int ltype, int ccltype)

int cv::connectedComponentsWithStats(InputArray image, OutputArray labels
    , OutputArray stats, OutputArray centroids, int connectivity, int
    ltype, int ccltype)
```

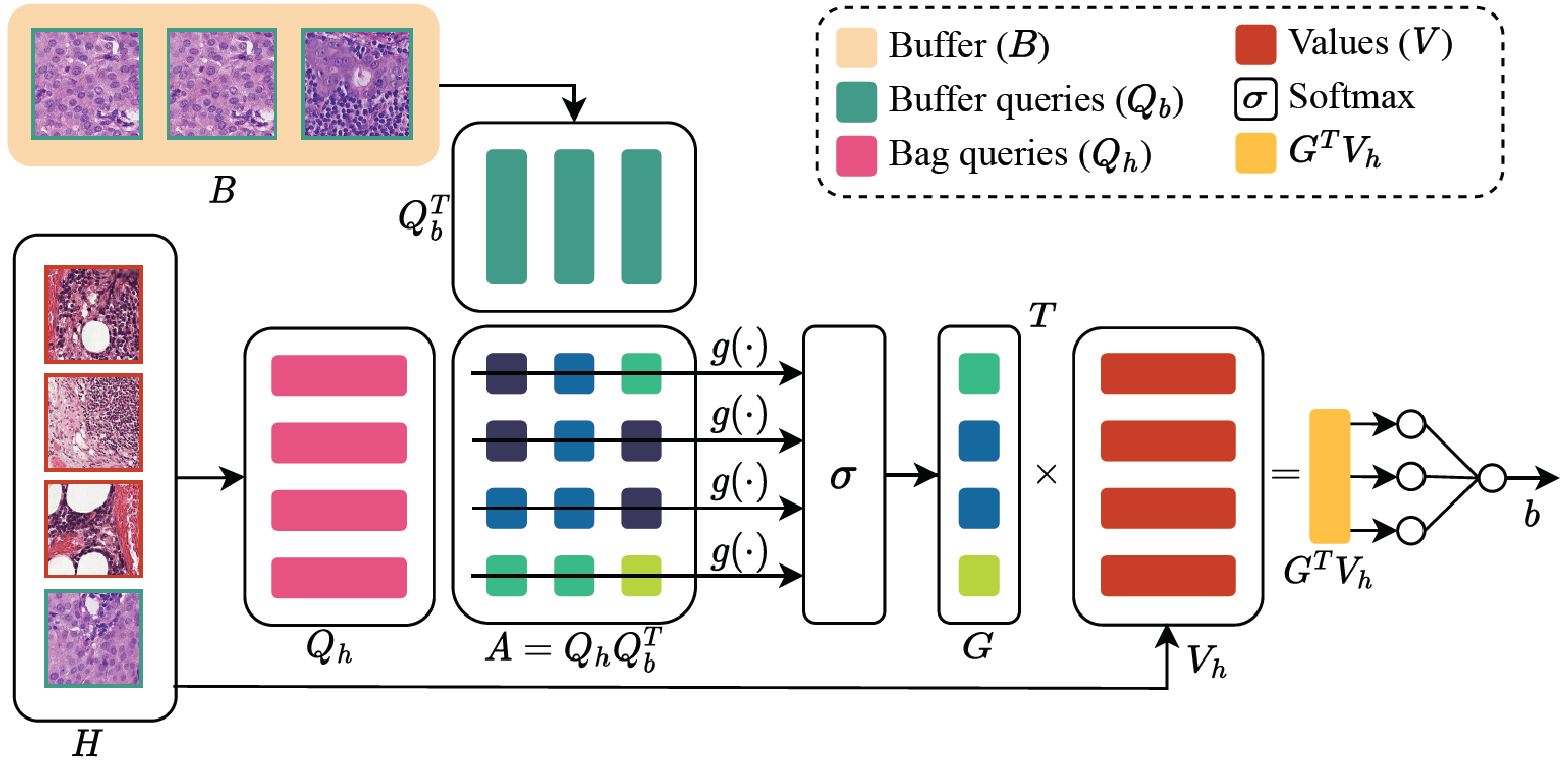
Listing 1.4. OpenCV C++ API for *connectedComponents* and *connectedComponents-WithStats* functions.

```
void cv::cuda::connectedComponents(InputArray image, OutputArray labels,
    int connectivity, int ltype, cv::cuda::
    ConnectedComponentsAlgorithmsTypes ccltype)
```

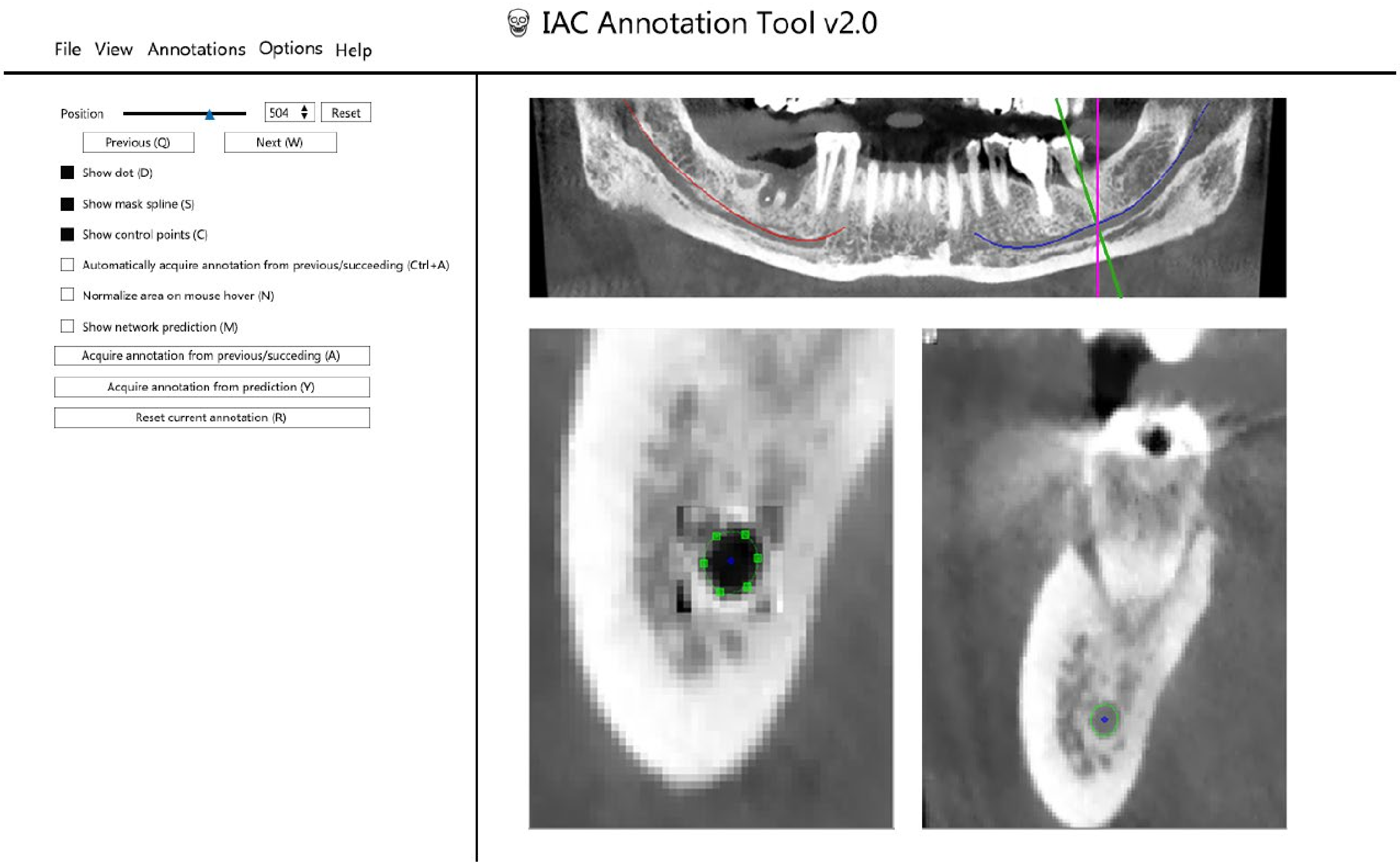
Listing 1.5. OpenCV C++ API for *connectedComponents* performed in CUDA.



Buffer (B)
Buffer queries (Q_b)
Bag queries (Q_h)
Values (V)
σ Softmax
$G^T V_h$
B
Q_b^T
H
Q_h
$A = Q_h Q_b^T$
$g(\cdot)$
$g(\cdot)$
$g(\cdot)$
$g(\cdot)$
σ
G
T
$\times$
V_h
$=$
$G^T V_h$
b



Buffer (B)
Buffer queries (Q_b)
Bag queries (Q_h)
Values (V)
σ Softmax
$G^T V_h$
B
Q_b^T
H
Q_h
$A = Q_h Q_b^T$
$g(\cdot)$
$g(\cdot)$
$g(\cdot)$
$g(\cdot)$
σ
G
T
$\times$
V_h
$=$
$G^T V_h$
b



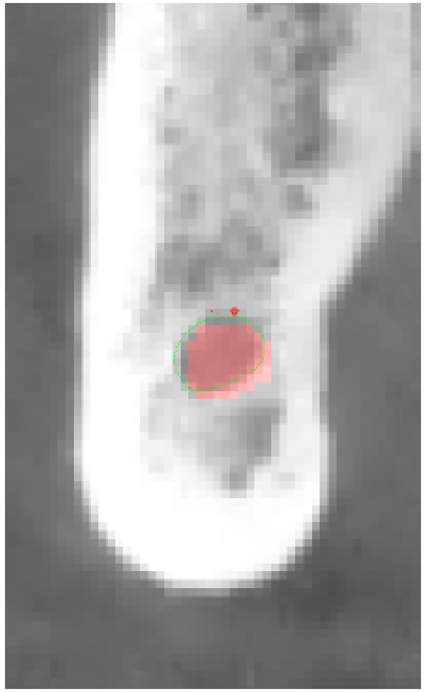
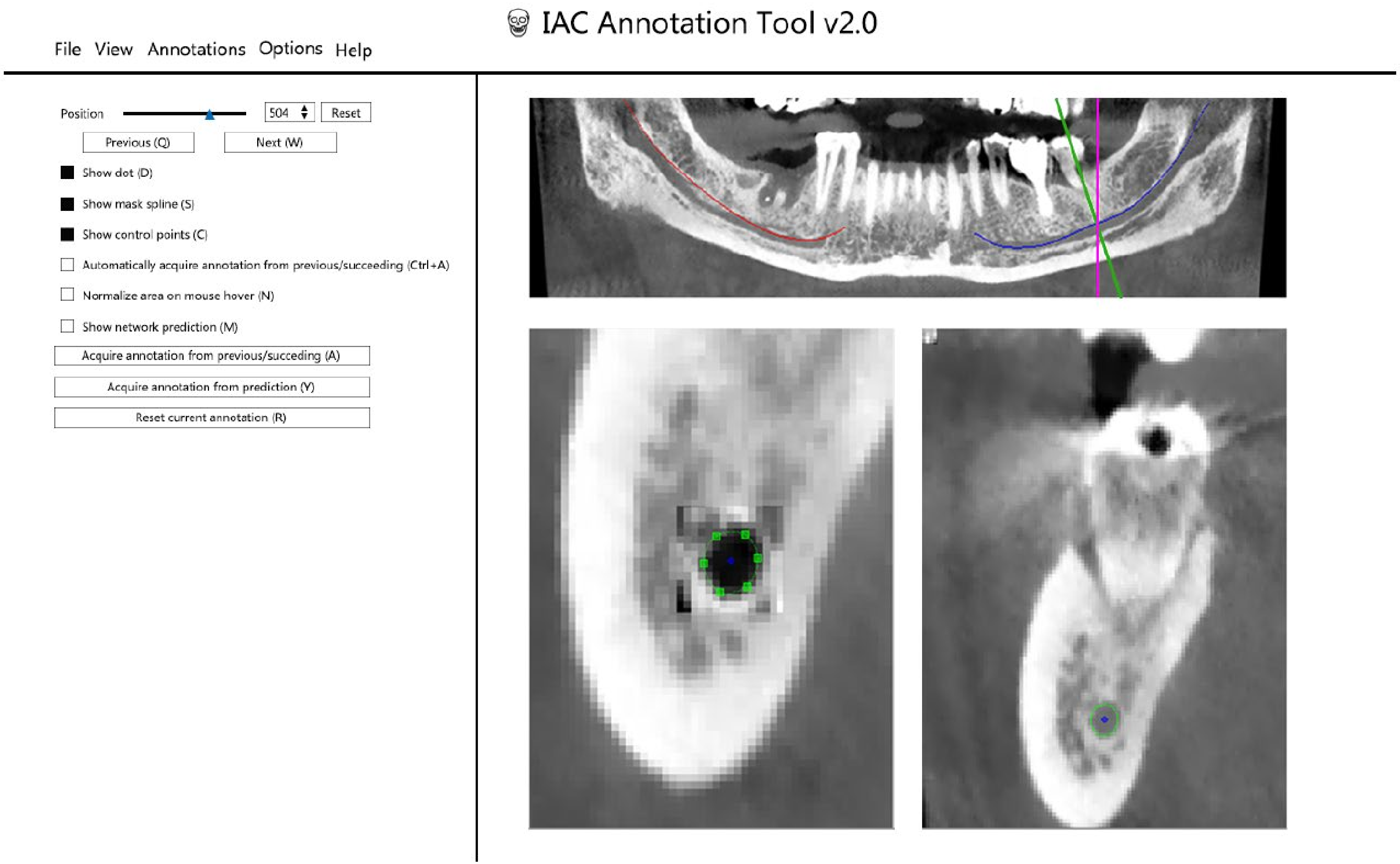


Fig. 4. CSV depicting the annotation performed by medical experts (in green), and the automatic prediction of the network (in red). The dark-red dot is the 2D sparse annotation.



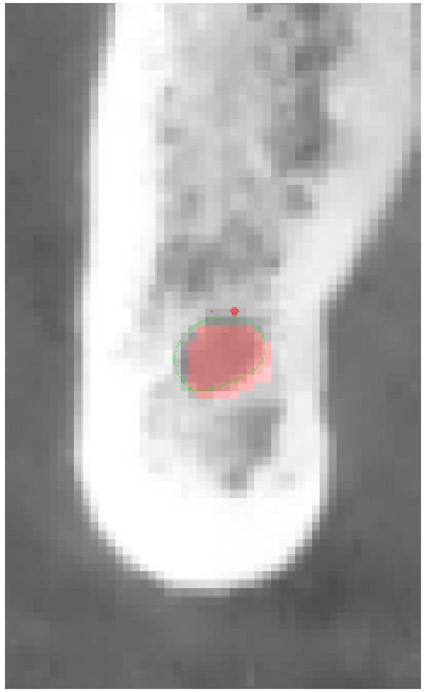
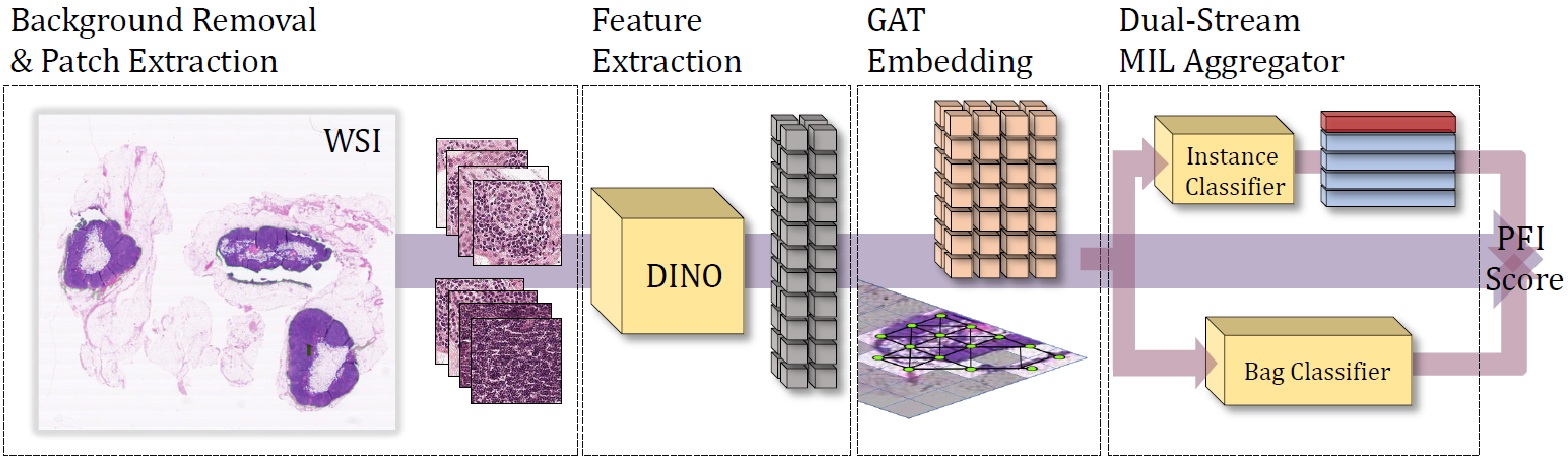
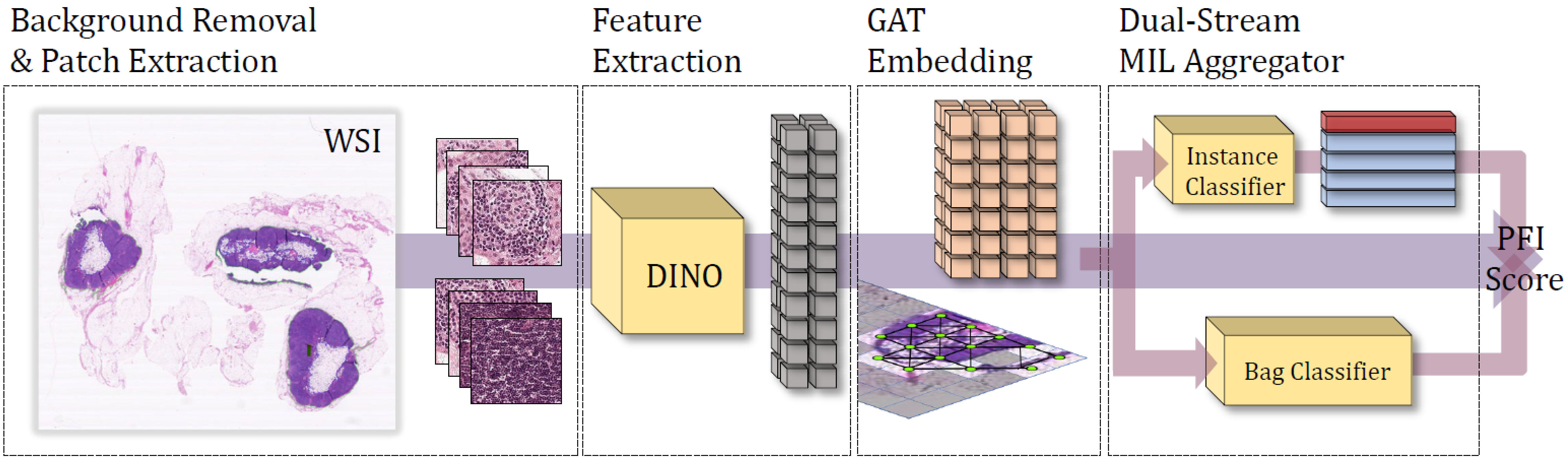
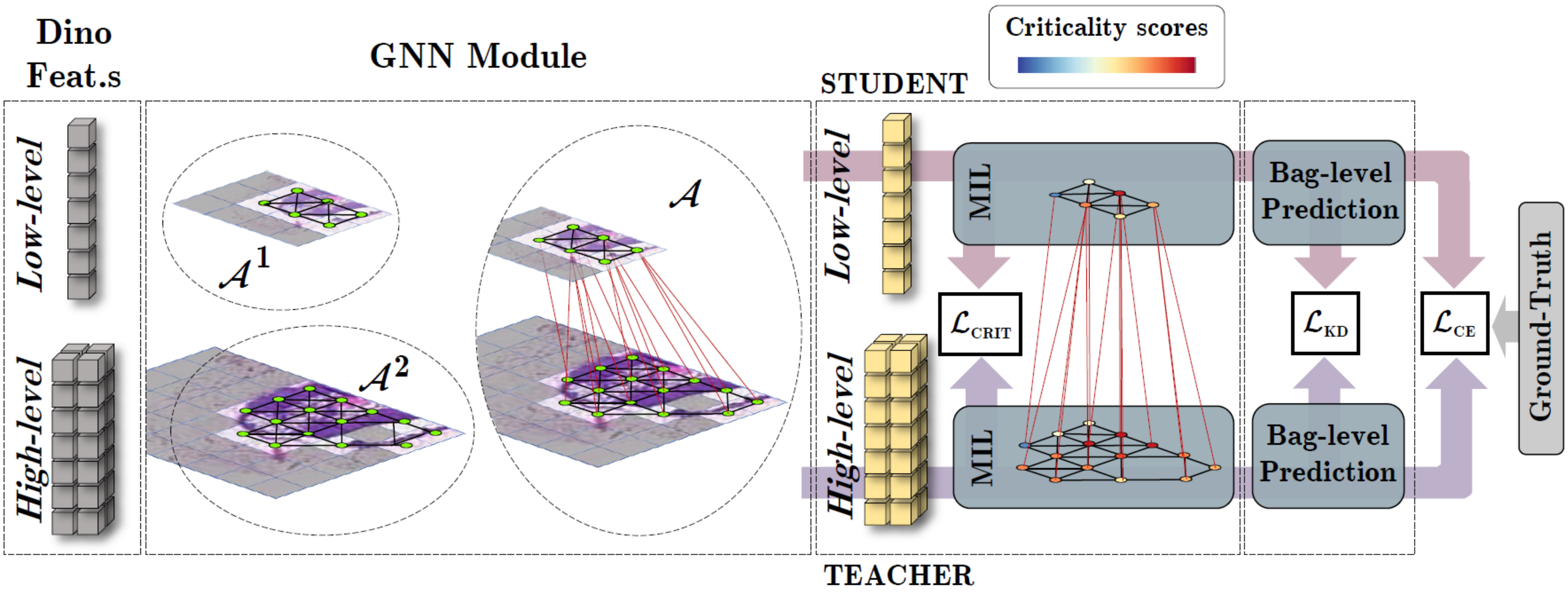


Fig. 4. CSV depicting the annotation performed by medical experts (in green), and the automatic prediction of the network (in red). The dark-red dot is the 2D sparse annotation.

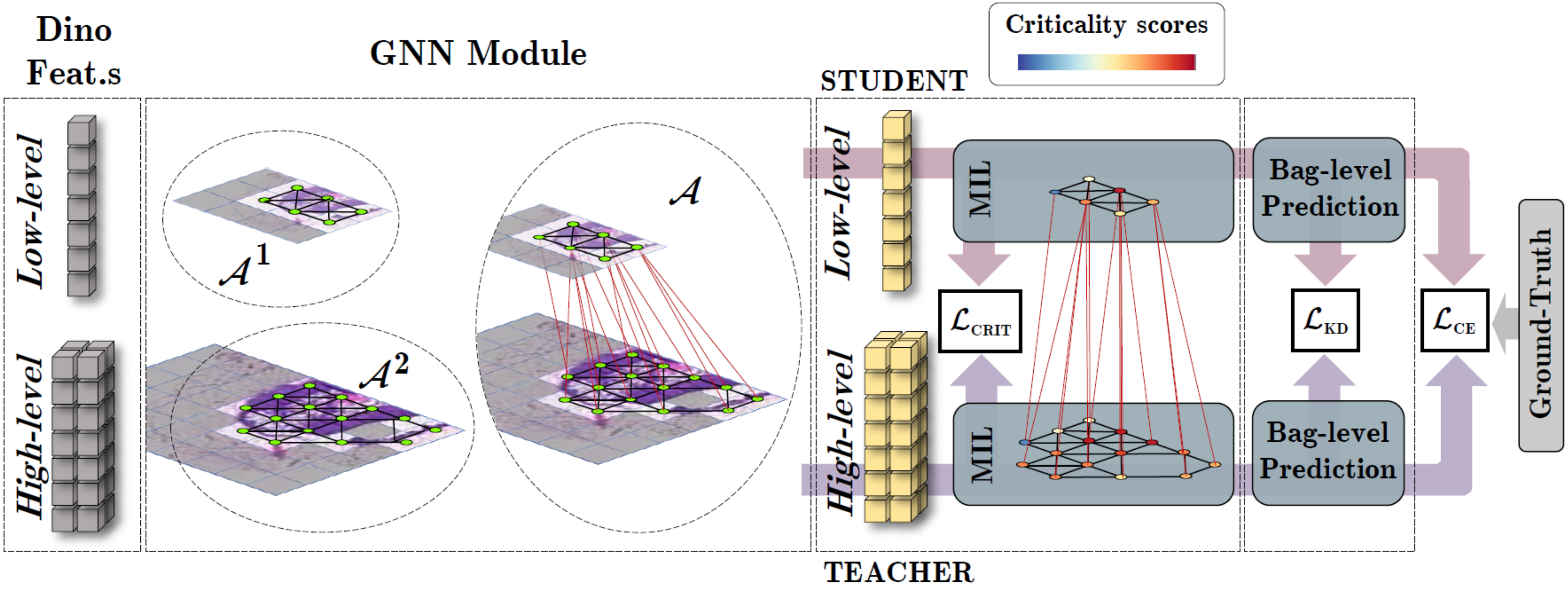

Background Removal
& Patch Extraction
Feature
Extraction
GAT
Embedding
Dual-Stream
MIL Aggregator
WSI
DINO
Instance
Classifier
Bag Classifier
PFI
Score



Background Removal
& Patch Extraction
Feature
Extraction
GAT
Embedding
Dual-Stream
MIL Aggregator
WSI
DINO
Instance
Classifier
Bag Classifier
PFI
Score



Dino
Feat.s
GNN Module
Low-level
High-level
$\mathcal{A}^1$
$\mathcal{A}^2$
$\mathcal{A}$
STUDENT
TEACHER
Criticality scores
MIL
MIL
Bag-level Prediction
Bag-level Prediction
$\mathcal{L}_{\text{CRIT}}$
$\mathcal{L}_{\text{KD}}$
$\mathcal{L}_{\text{CE}}$
Ground-Truth



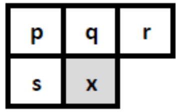
Dino Feat.s
Low-level
High-level
GNN Module
$\mathcal{A}^1$
$\mathcal{A}^2$
$\mathcal{A}$
STUDENT
TEACHER
Criticality scores
MIL
MIL
$\mathcal{L}_{\text{CRIT}}$
Bag-level Prediction
Bag-level Prediction
$\mathcal{L}_{\text{KD}}$
$\mathcal{L}_{\text{CE}}$
Ground-Truth

MICCAI 2023
Vancouver
CANADA

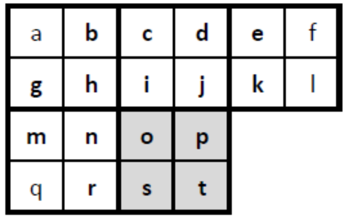
TOOTH
FAIRY

MICCAI 2023
Vancouver
CANADA

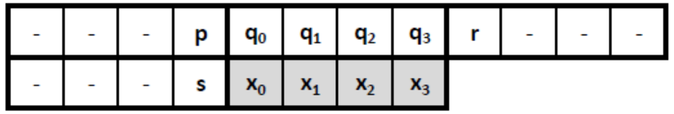
TOOTH
FAIRY



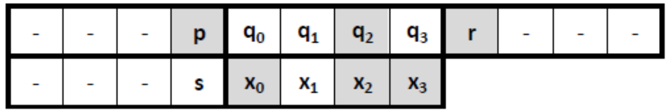
(a) Rosenfeld



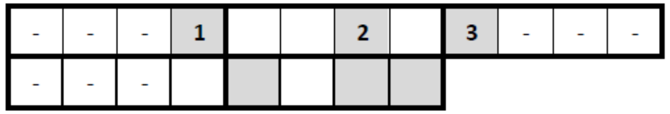
(b) Grana



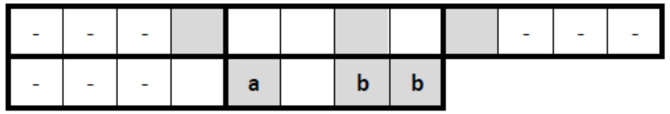
(c) Rosenfeld4



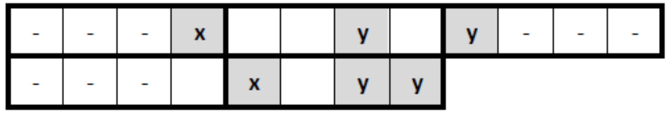
(a) Configuration



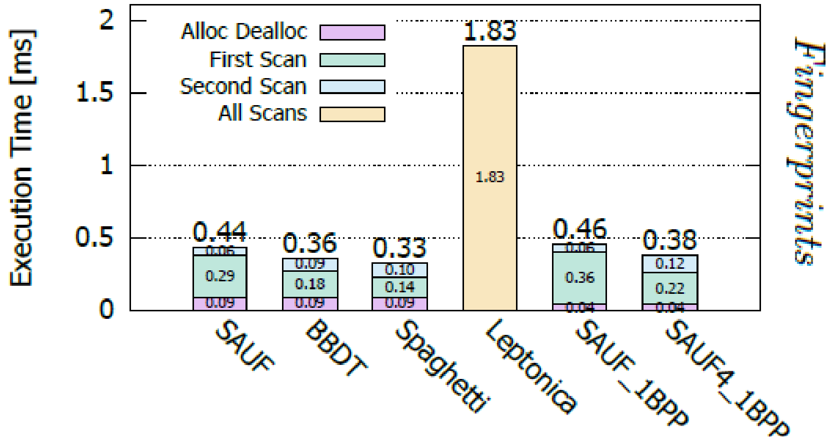
(b) Outer CCs

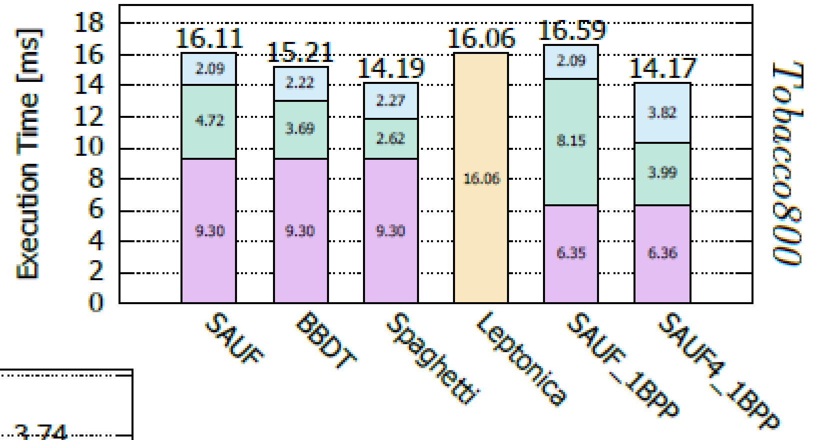


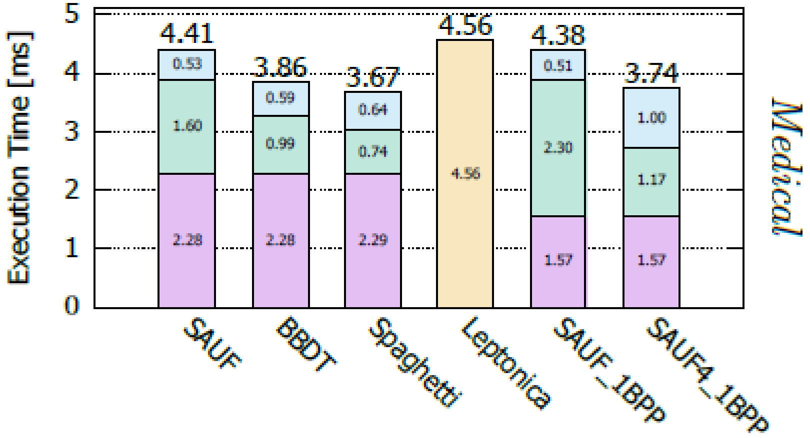
(c) Inner CCs

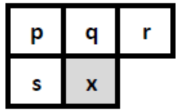


(d) Global CCs

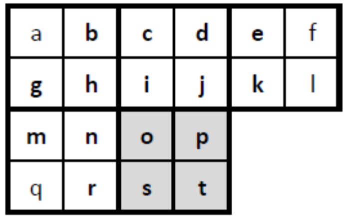




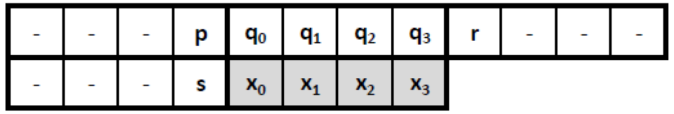




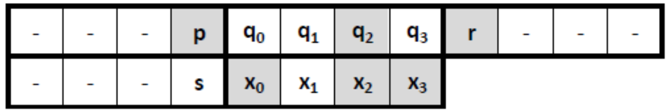
(a) Rosenfeld



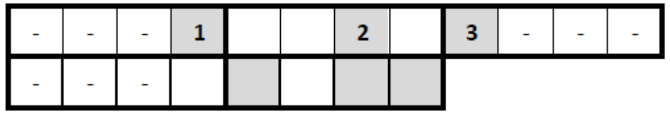
(b) Grana



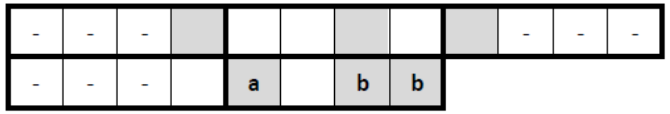
(c) Rosenfeld4



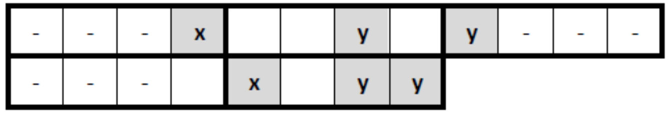
(a) Configuration



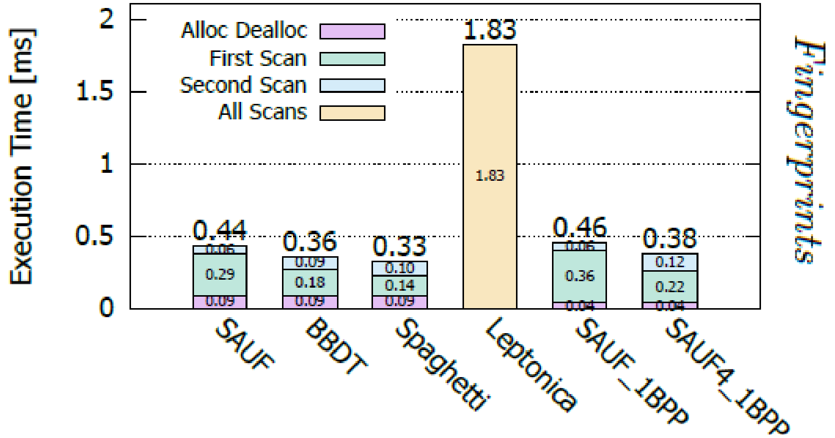
(b) Outer CCs

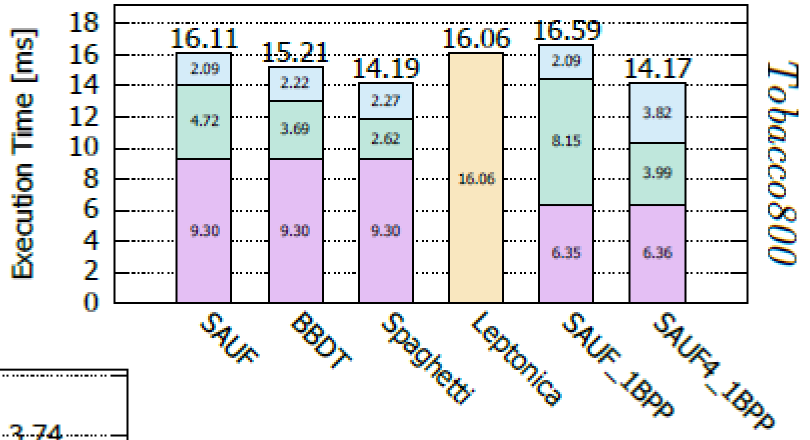


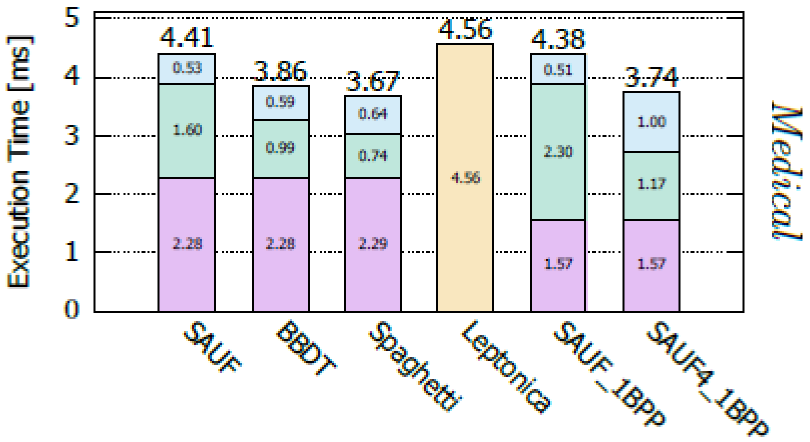
(c) Inner CCs

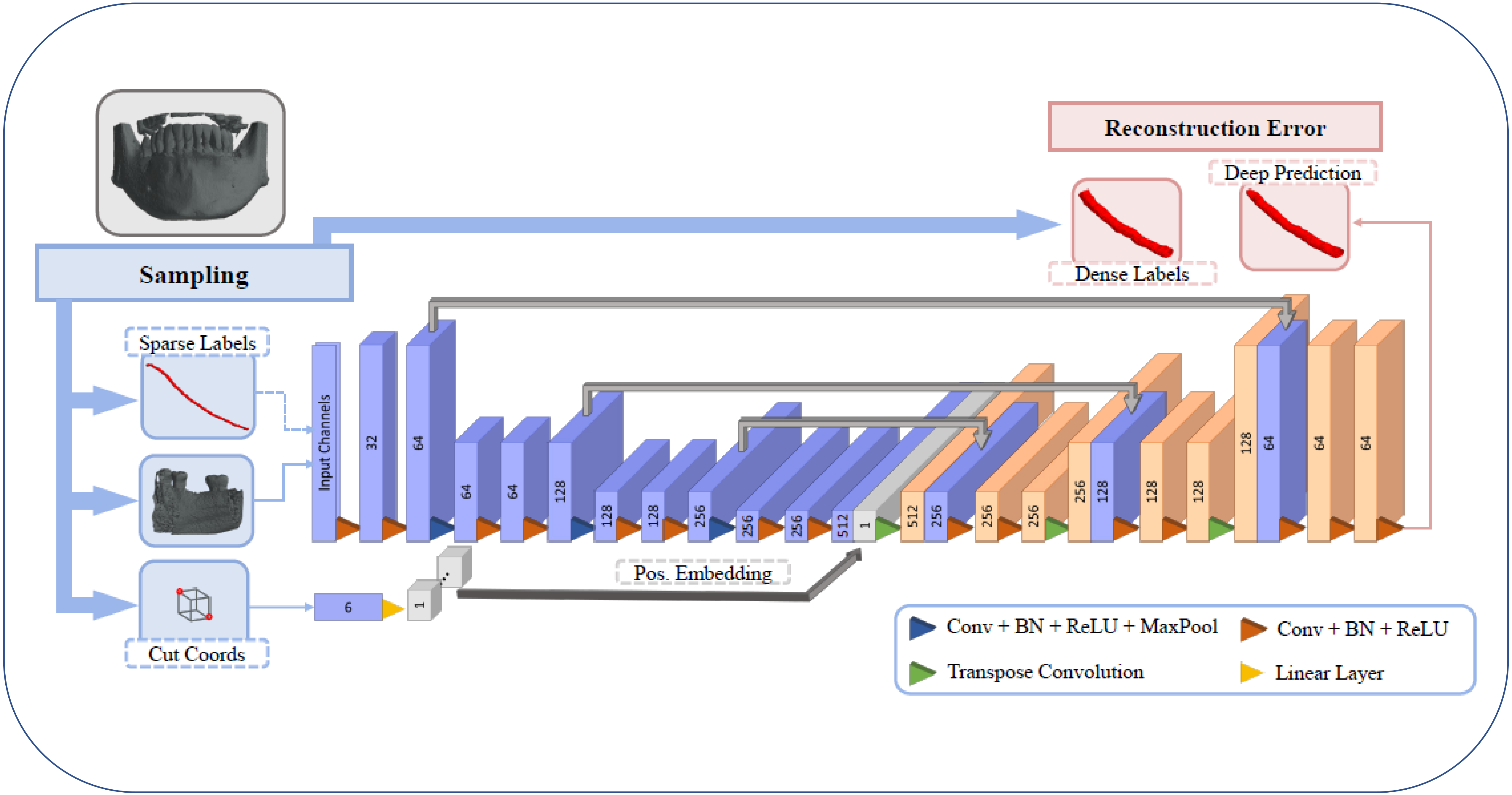


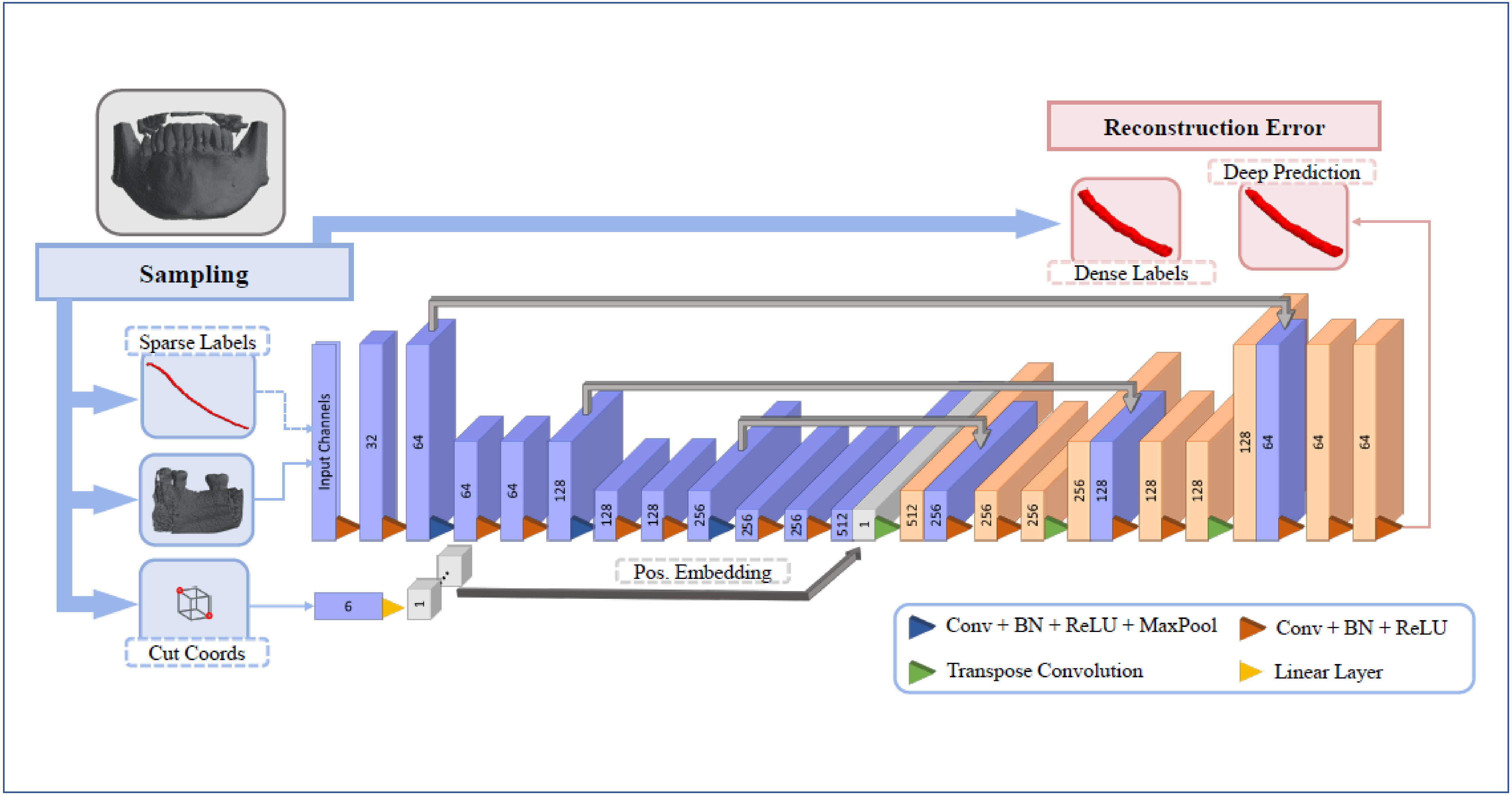
(d) Global CCs



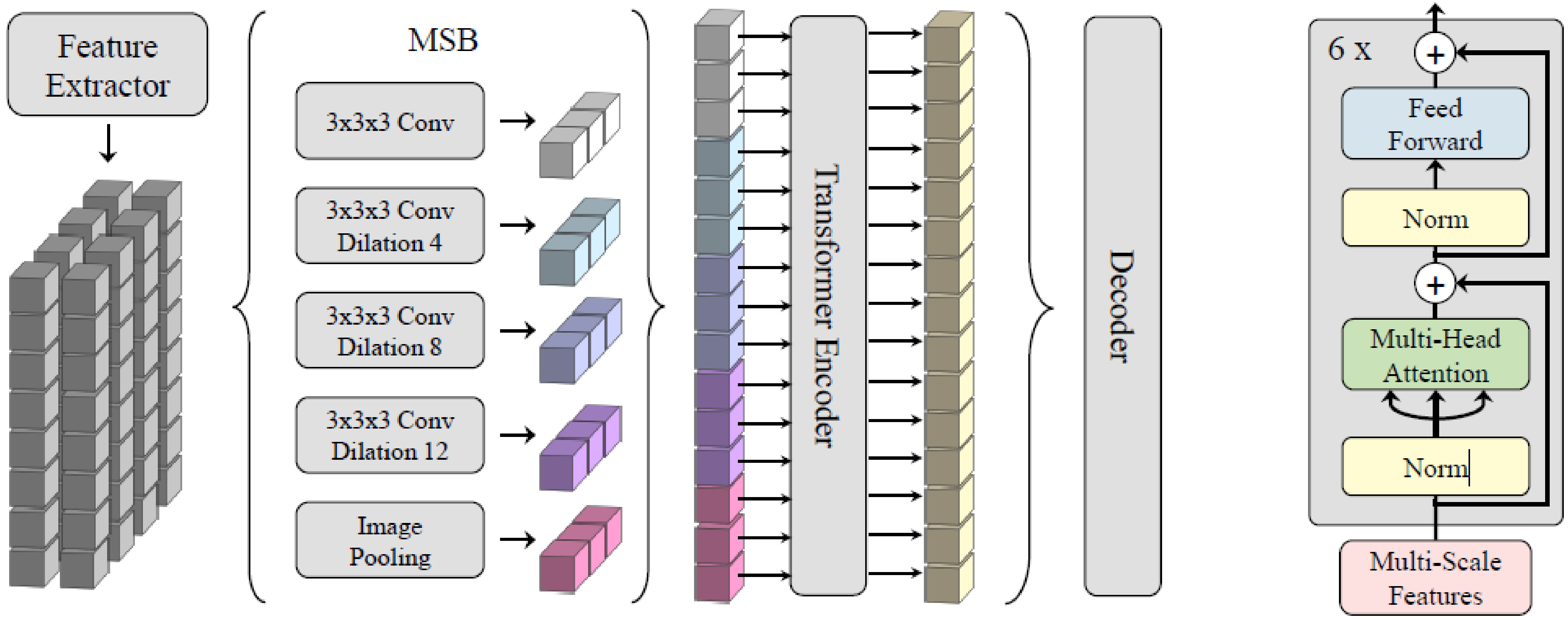




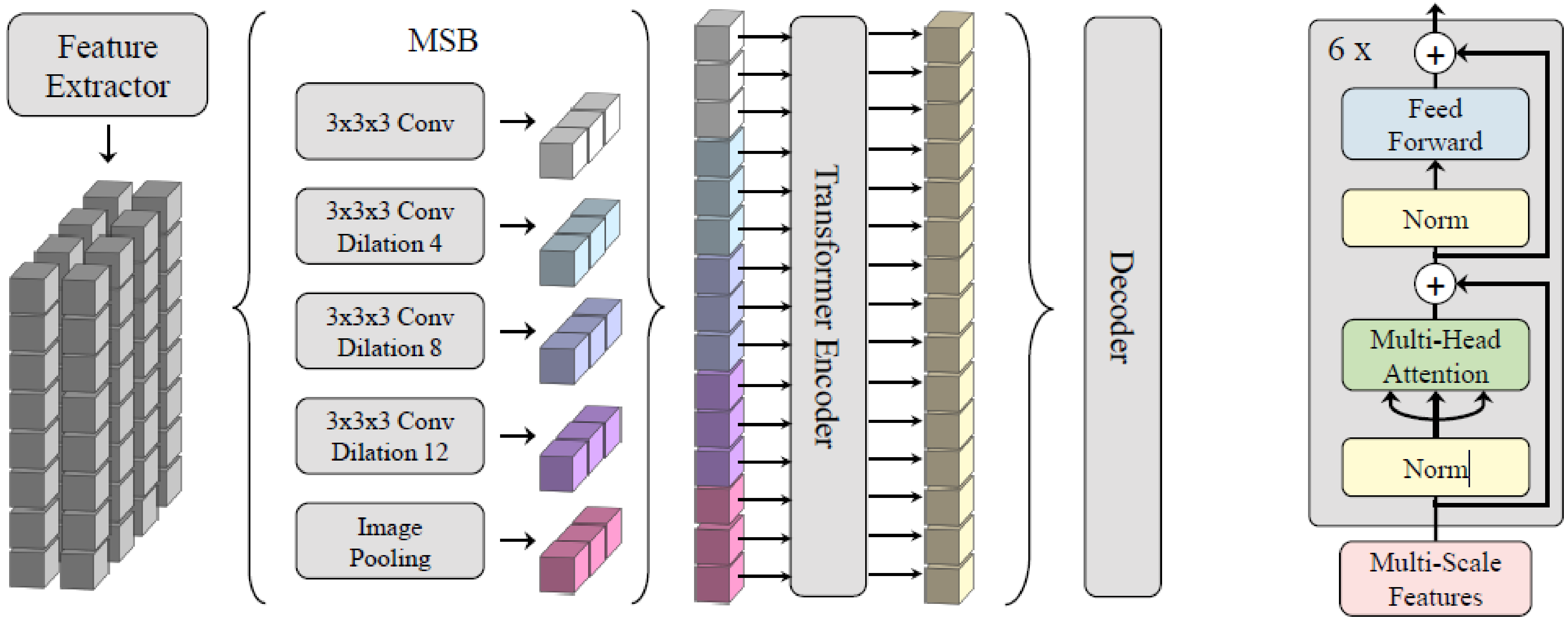

Reconstruction Error
Deep Prediction
Dense Labels
Sampling
Sparse Labels
Cut Coords
Input Channels
32
64
64
64
128
128
128
256
256
256
512
1
512
256
256
256
256
128
128
128
128
64
64
64
6
1
Pos. Embedding
Conv + BN + ReLU + MaxPool
Conv + BN + ReLU
Transpose Convolution
Linear Layer



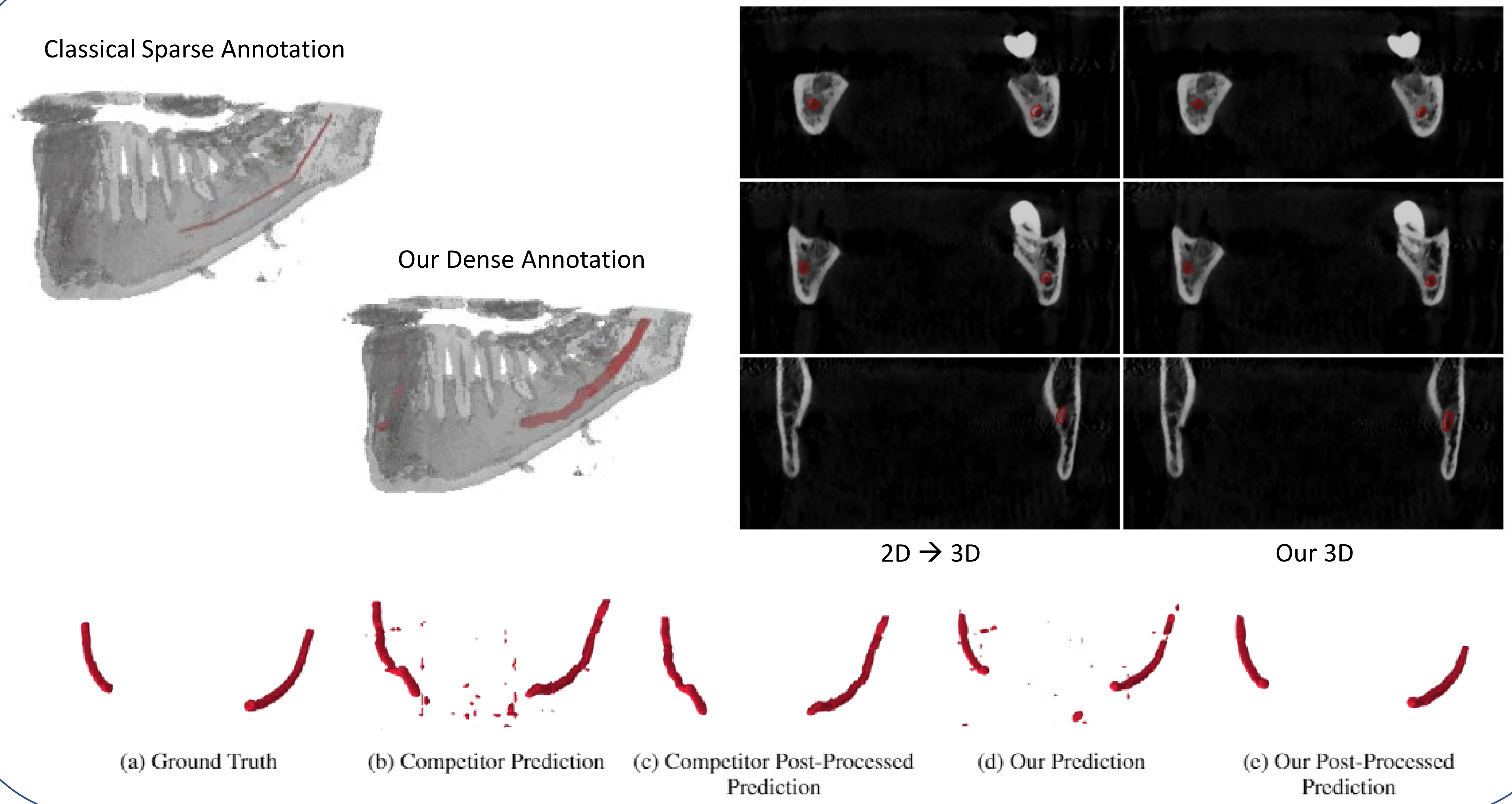
Reconstruction Error
Deep Prediction
Dense Labels
Sampling
Sparse Labels
Cut Coords
Input Channels
32
64
64
64
128
128
128
256
256
256
512
1
512
256
256
256
256
128
128
128
128
64
64
64
6
1
Pos. Embedding
Conv + BN + ReLU + MaxPool
Conv + BN + ReLU
Transpose Convolution
Linear Layer

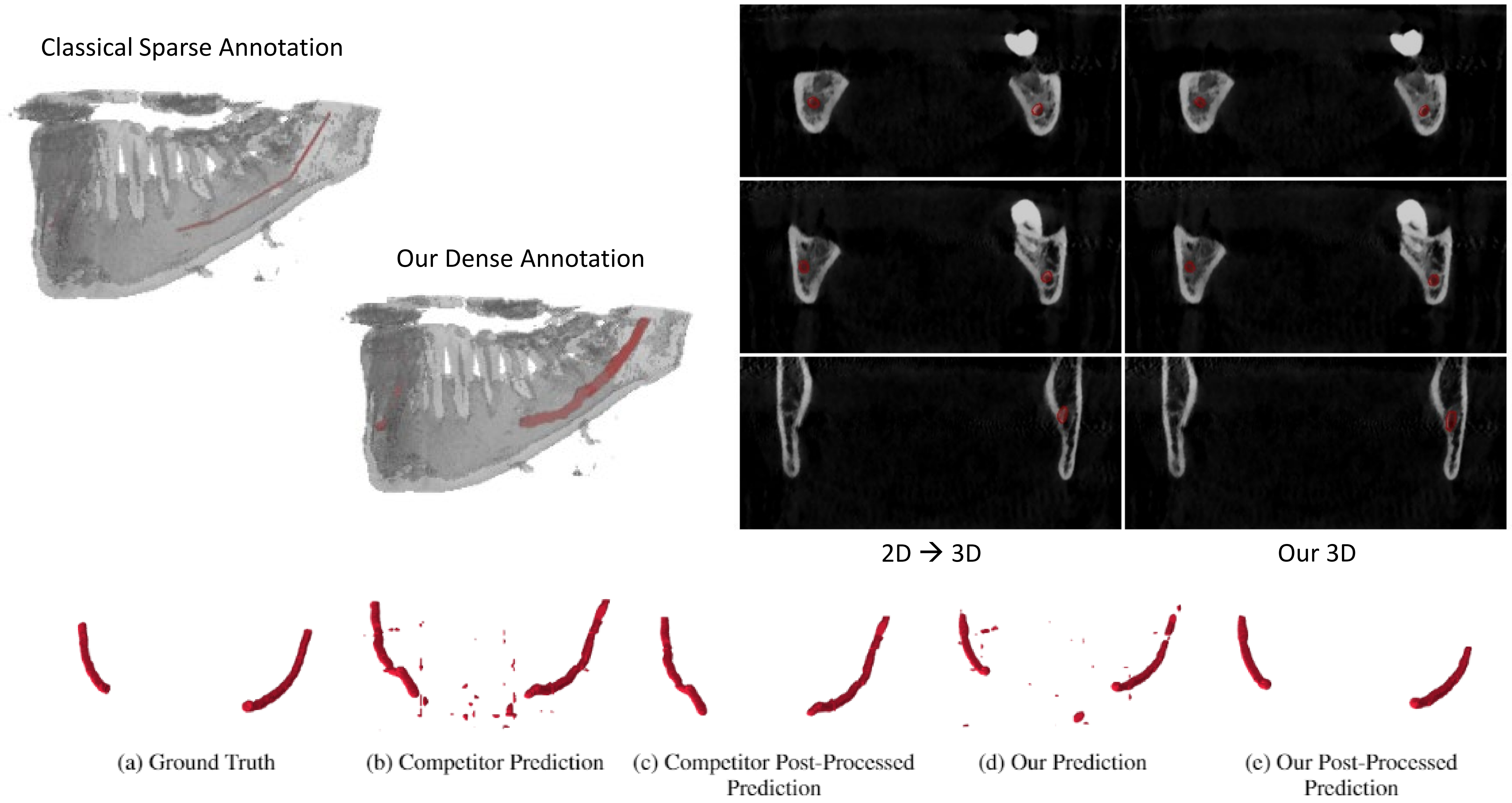


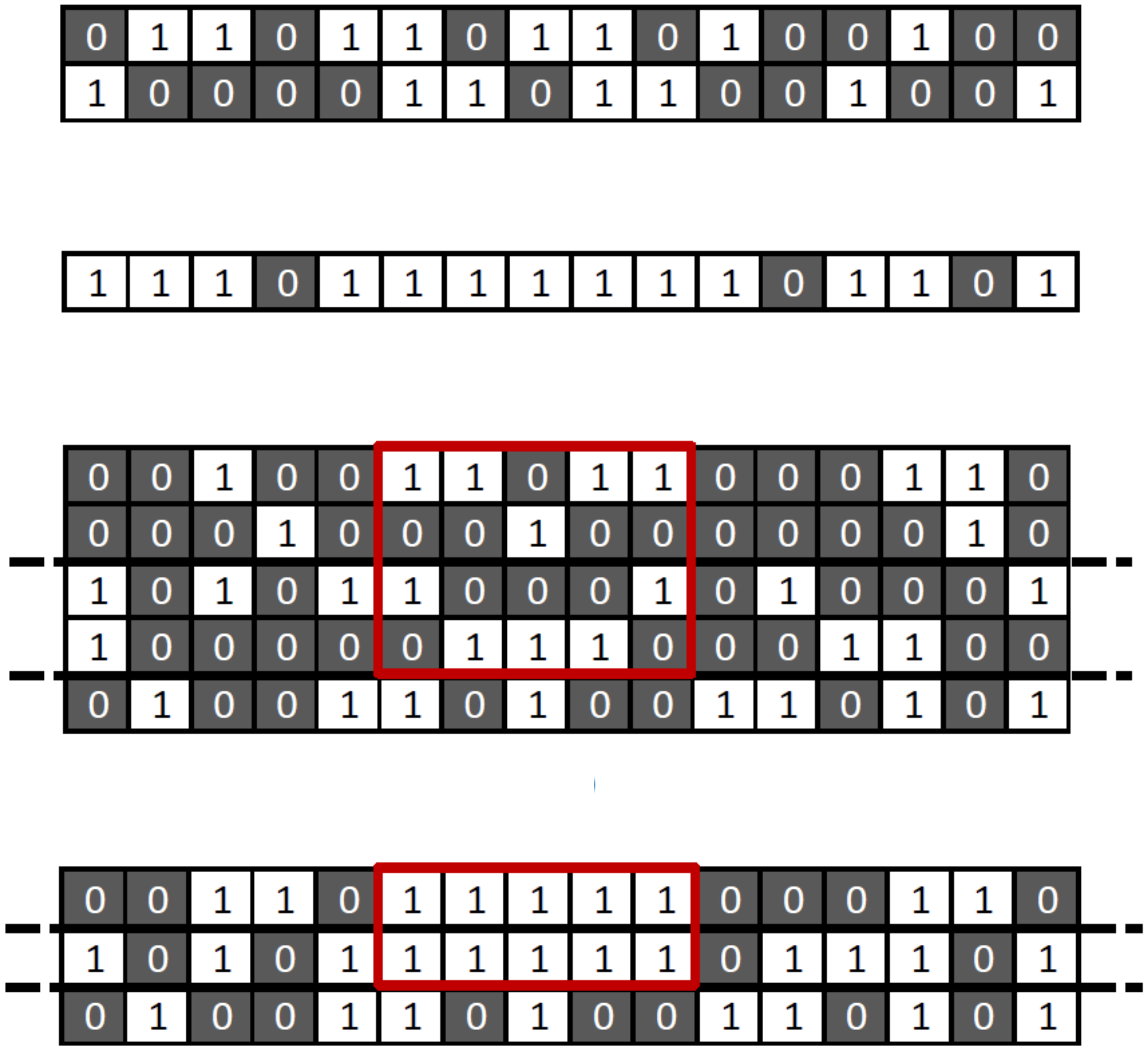
Feature Extractor
MSB
3x3x3 Conv
3x3x3 Conv Dilation 4
3x3x3 Conv Dilation 8
3x3x3 Conv Dilation 12
Image Pooling
Transformer Encoder
Decoder
6 x
Feed Forward
Norm
Multi-Head Attention
Norm
Multi-Scale Features

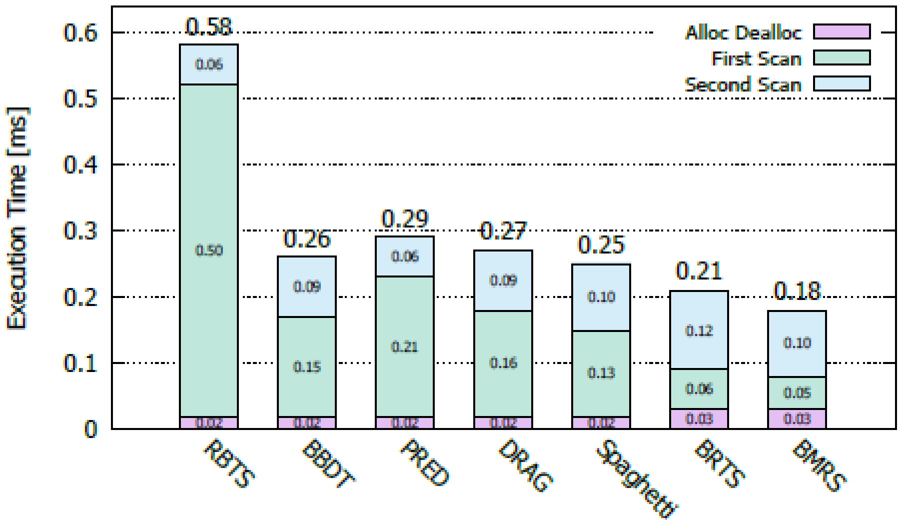


Feature Extractor
MSB
3x3x3 Conv
3x3x3 Conv Dilation 4
3x3x3 Conv Dilation 8
3x3x3 Conv Dilation 12
Image Pooling
Transformer Encoder
Decoder
6 x
Feed Forward
Norm
Multi-Head Attention
Norm
Multi-Scale Features


Classical Sparse Annotation
Our Dense Annotation
2D → 3D
Our 3D
(a) Ground Truth
(b) Competitor Prediction
(c) Competitor Post-Processed Prediction
(d) Our Prediction
(e) Our Post-Processed Prediction

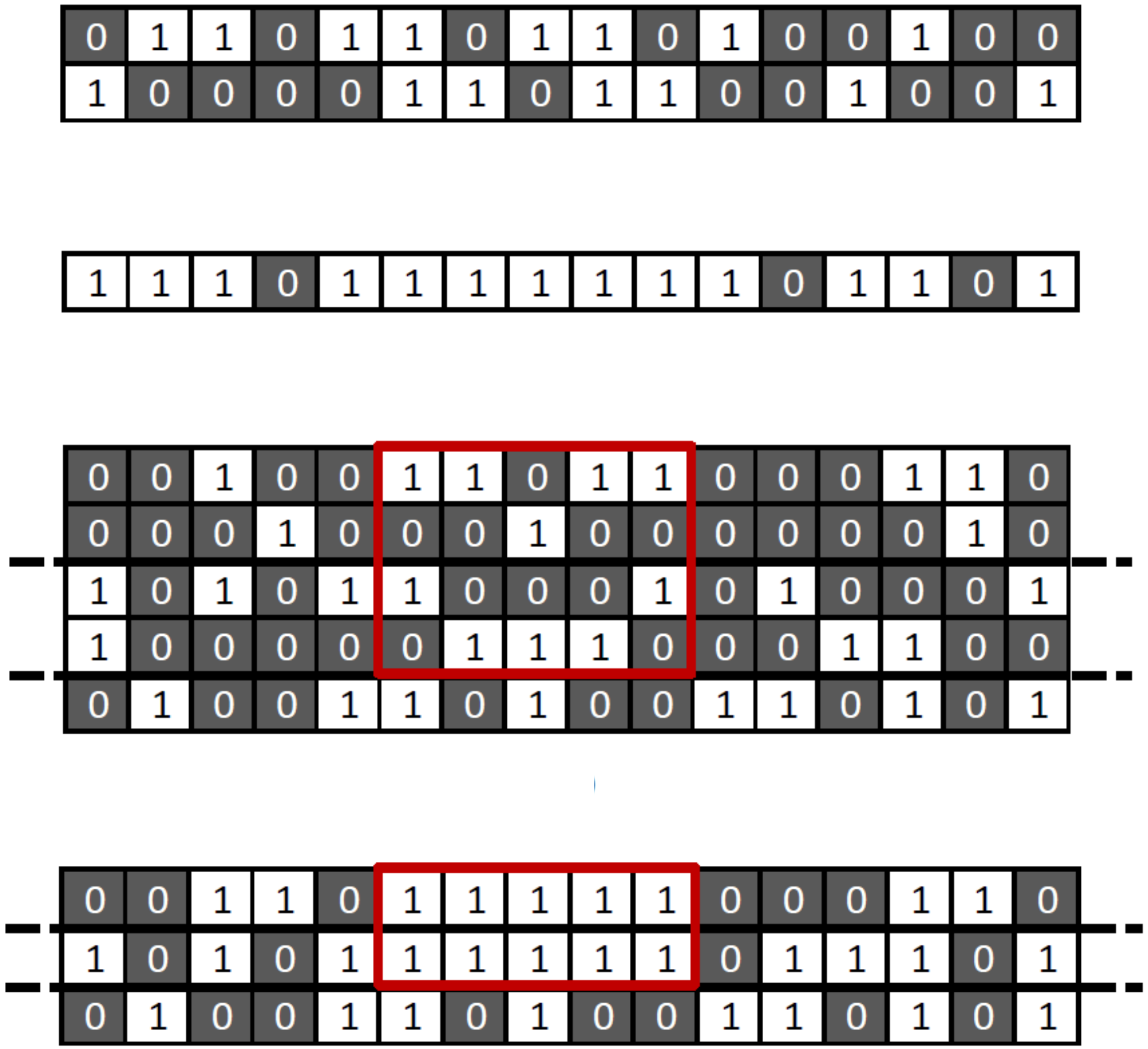

Classical Sparse Annotation
Our Dense Annotation
2D → 3D
Our 3D
(a) Ground Truth
(b) Competitor Prediction
(c) Competitor Post-Processed Prediction
(d) Our Prediction
(e) Our Post-Processed Prediction

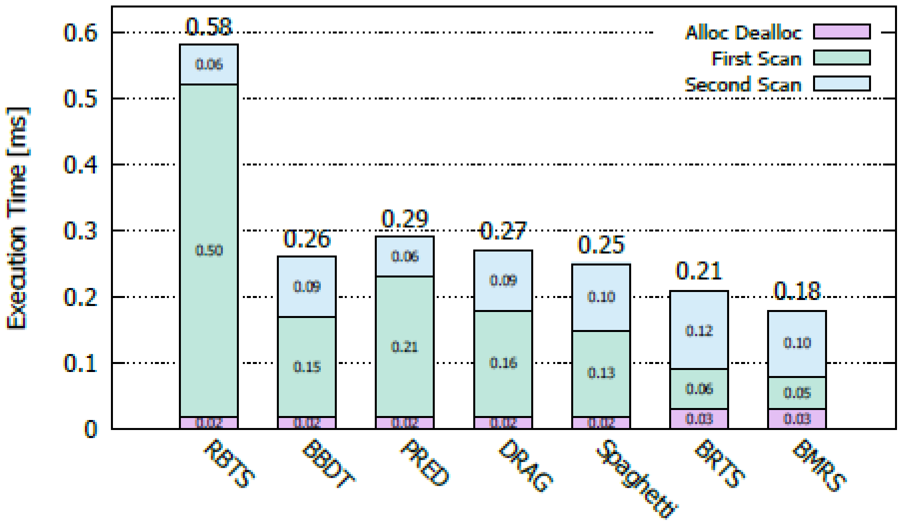




(a) *Fingerprints*

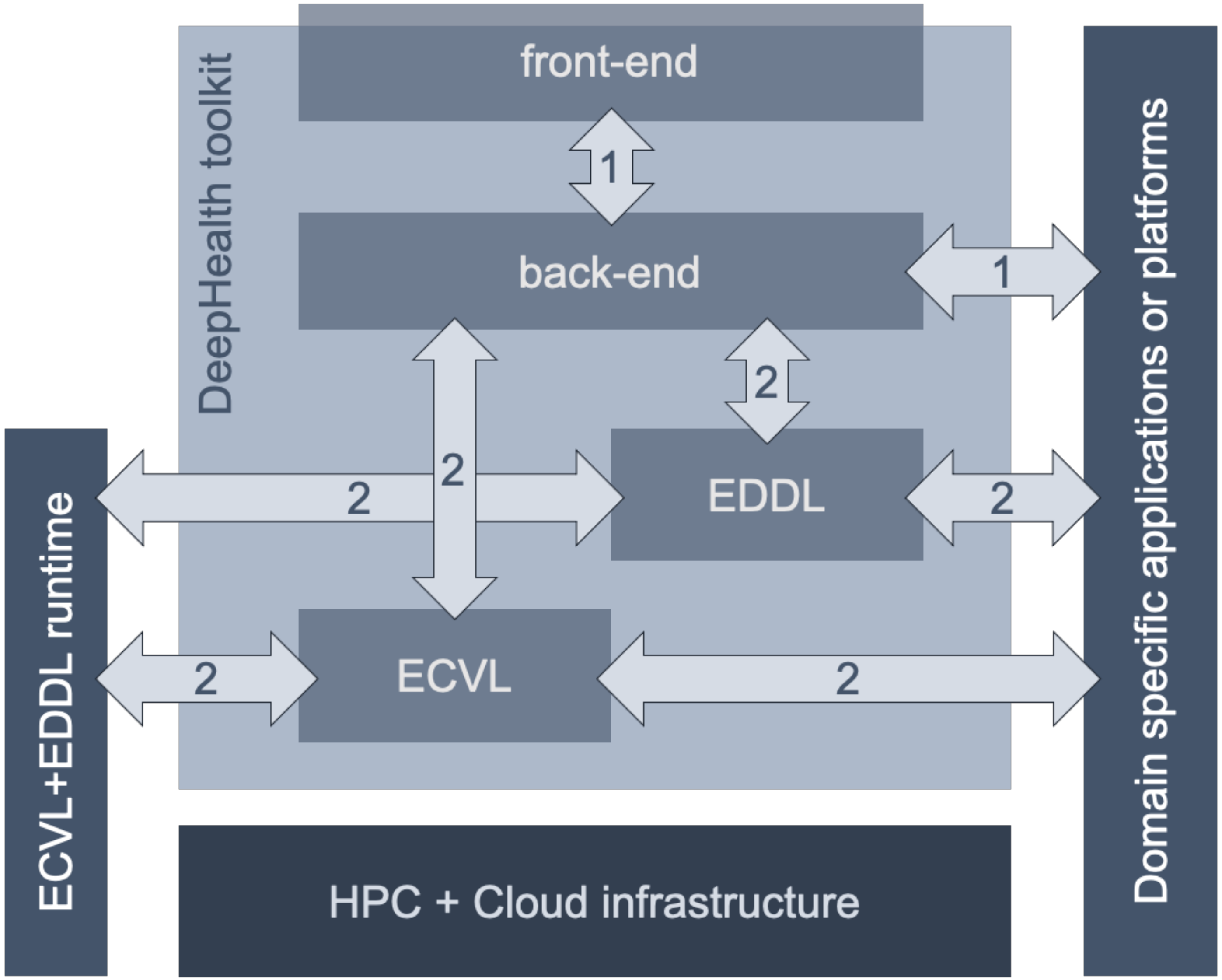
(b) *3dpes*





(a) *Fingerprints*

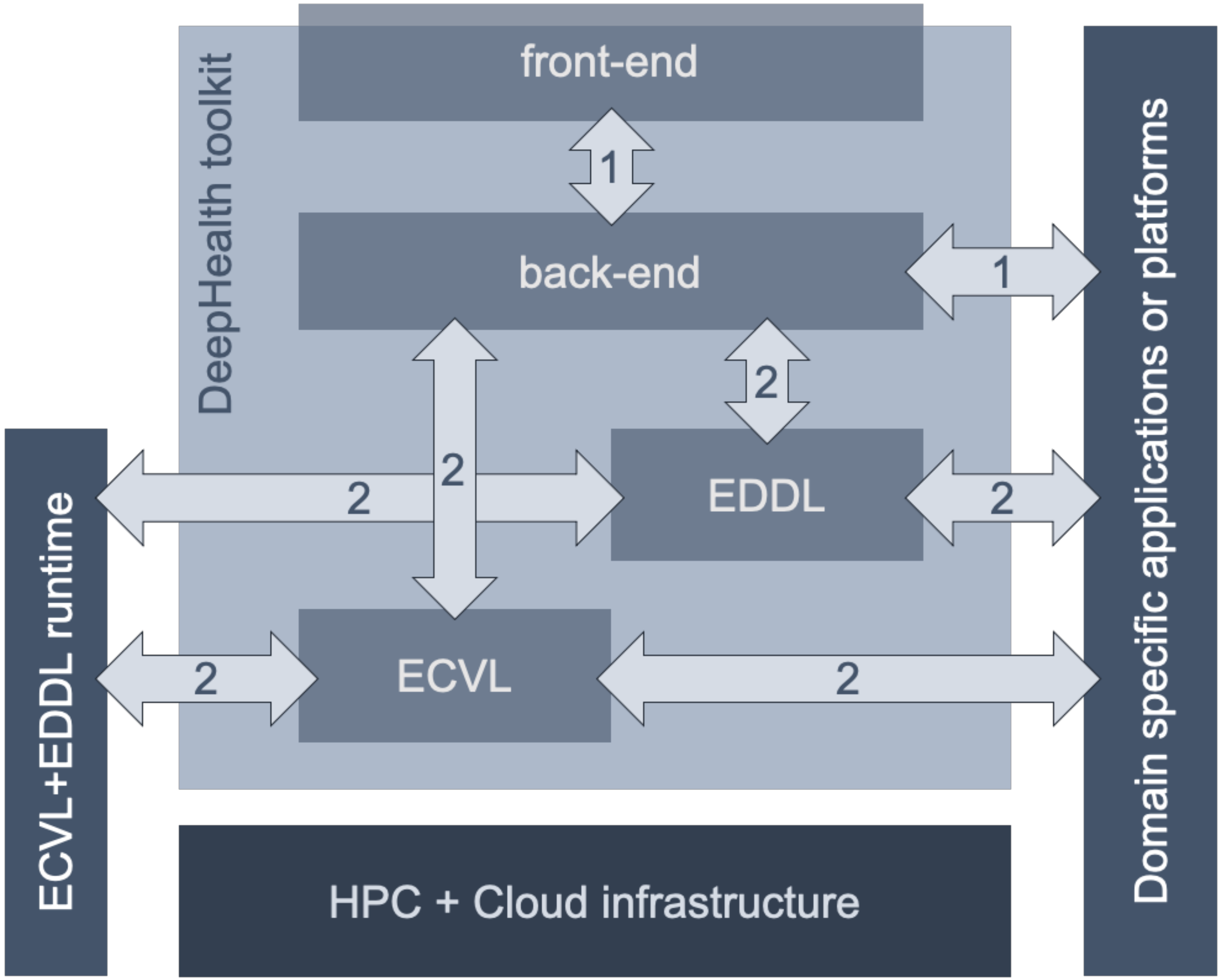
(b) *3dpes*



DEEPHEALTH

Model	Accuracy / Time	Tensorflow		Pytorch		EDDL	
		No BN	BN	No BN	BN	No BN	BN
VGG16	Test accuracy	77.4 %	71.7 %	77.9 %	/ 76.2 %	74.6 %	76.4 %
	GPU time per epoch	62 s	68 s	72 s	77 s	146 s	204 s
	CPU time per epoch	1313 s	1375 s	887 s	956 s	3107 s	2846 s
VGG19	Test accuracy	66.0 %	59.9 %	65.5 %	59.7 %	68.2 %	61.0 %
	GPU time per epoch	76 s	81 s	120 s	126 s	190 s	260 s
	CPU time per epoch	1703 s	1809 s	1262 s	1352 s	3872 s	3838 s
RestNet18	Test accuracy	67.6 %	64.0 %	66.4 %	65.7 %	67.3 %	64.8 %
	GPU time per epoch	25 s	26 s	59 s	60 s	36 s	49 s
	CPU time per epoch	1234 s	1244 s	456 s	485 s	932 s	1207 s
ResNet34	Test accuracy	66.6 %	66.4 %	67.8 %	65.5 %	66.1 %	60.4 %
	GPU time per epoch	44 s	46 s	97 s	101 s	65 s	89 s
	CPU time per epoch	2125 s	2140 s	834 s	895 s	1674 s	2119 s
ResNet50	Test accuracy	68.4 %	61.3 %	68.1 %	63.1 %	66.4 %	61.9 %
	GPU time per epoch	47 s	52 s	84 s	92 s	75 s	132 s
	CPU time per epoch	1995 s	2044 s	706 s	835 s	1684 s	2622 s

Benchmark to compare EDDL with TensorFlow and PyTorch using Cifar10 with and without Batch Normalization.



DEEPHEALTH

Model	Accuracy / Time	Tensorflow		Pytorch		EDDL	
		No BN	BN	No BN	BN	No BN	BN
VGG16	Test accuracy	77.4 %	71.7 %	77.9 %	/ 76.2 %	74.6 %	76.4 %
	GPU time per epoch	62 s	68 s	72 s	77 s	146 s	204 s
	CPU time per epoch	1313 s	1375 s	887 s	956 s	3107 s	2846 s
VGG19	Test accuracy	66.0 %	59.9 %	65.5 %	59.7 %	68.2 %	61.0 %
	GPU time per epoch	76 s	81 s	120 s	126 s	190 s	260 s
	CPU time per epoch	1703 s	1809 s	1262 s	1352 s	3872 s	3838 s
RestNet18	Test accuracy	67.6 %	64.0 %	66.4 %	65.7 %	67.3 %	64.8 %
	GPU time per epoch	25 s	26 s	59 s	60 s	36 s	49 s
	CPU time per epoch	1234 s	1244 s	456 s	485 s	932 s	1207 s
ResNet34	Test accuracy	66.6 %	66.4 %	67.8 %	65.5 %	66.1 %	60.4 %
	GPU time per epoch	44 s	46 s	97 s	101 s	65 s	89 s
	CPU time per epoch	2125 s	2140 s	834 s	895 s	1674 s	2119 s
ResNet50	Test accuracy	68.4 %	61.3 %	68.1 %	63.1 %	66.4 %	61.9 %
	GPU time per epoch	47 s	52 s	84 s	92 s	75 s	132 s
	CPU time per epoch	1995 s	2044 s	706 s	835 s	1684 s	2622 s

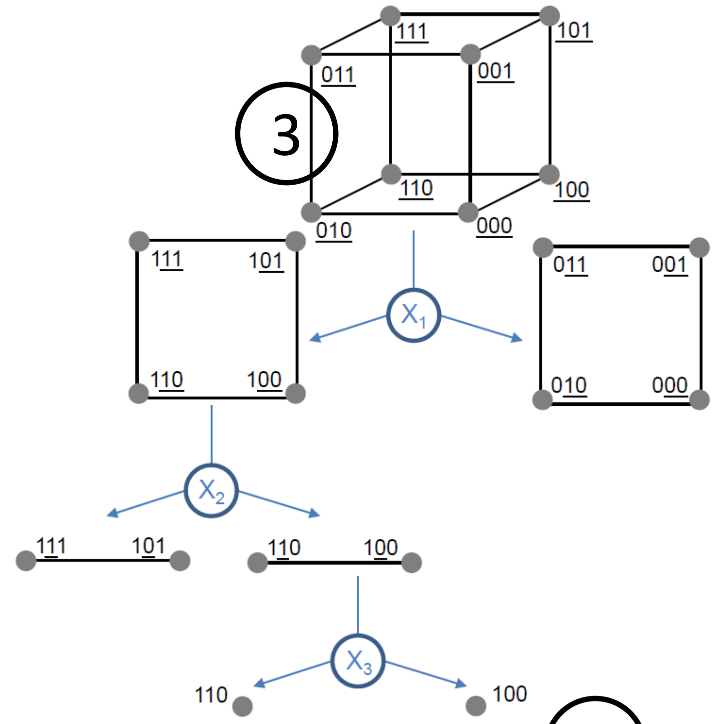
Benchmark to compare EDDL with TensorFlow and PyTorch using Cifar10 with and without Batch Normalization.

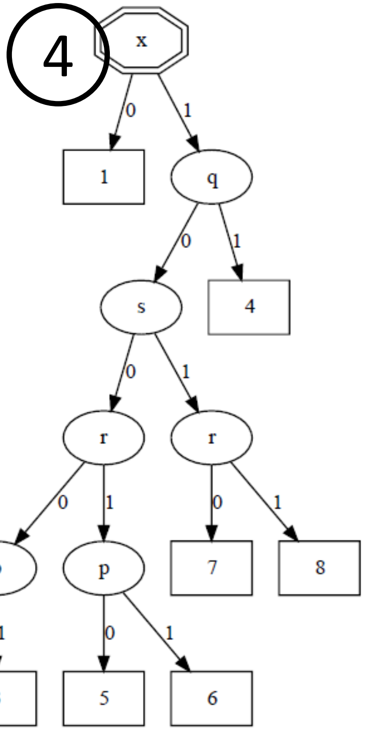
1

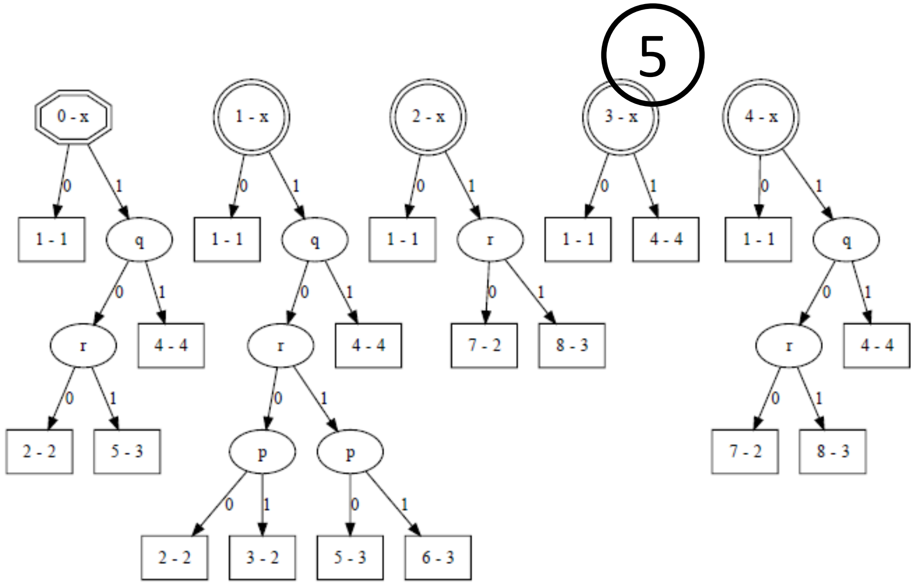
```
pixel_set:
  pixels:
      - {name: p, coords: [-1, -1]}
      - {name: x, coords: [ 0, 0]}
      - ...
  shifts: [1, 1]
conditions: [p, q, r, s, x]
actions: [nothing, x<-newlabel, x<-p, x<-q, x<-r, x
     <-p+r, x<-s, x<-r+s]
rules: [[1], [1], ..., [2], [3], [4], [3, 4], [5],
     ..., [3, 4, 5, 7]]
```

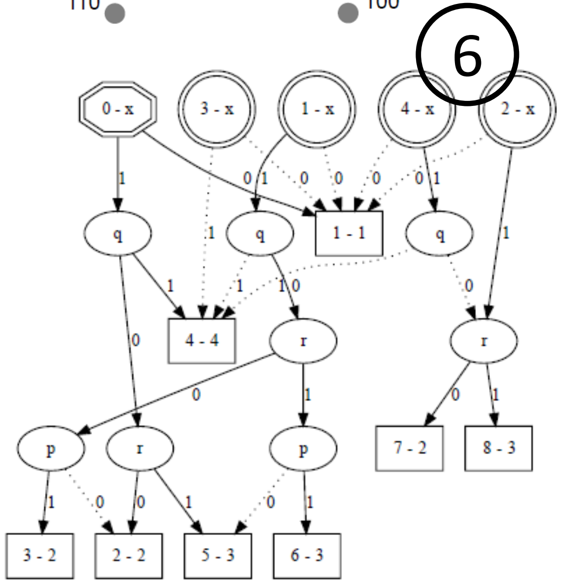
2

Conditions	x	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	condition outcomes
	p	-	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	
	q	-	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1	
	r	-	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1	
	s	-	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	
Actions	no op.	1																	action entries
	new		1																
	p			1				1		1				1	1			1	
	q				1			1			1	1		1	1		1	1	
	r					1					1			1			1	1	
	s						1			1		1			1		1	1	
	p + r								1							1			
	r + s												1			1			







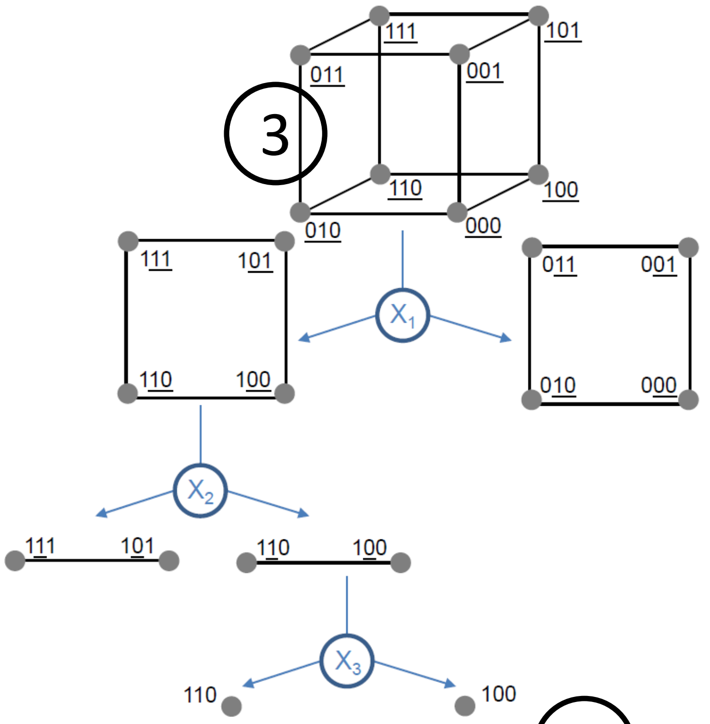


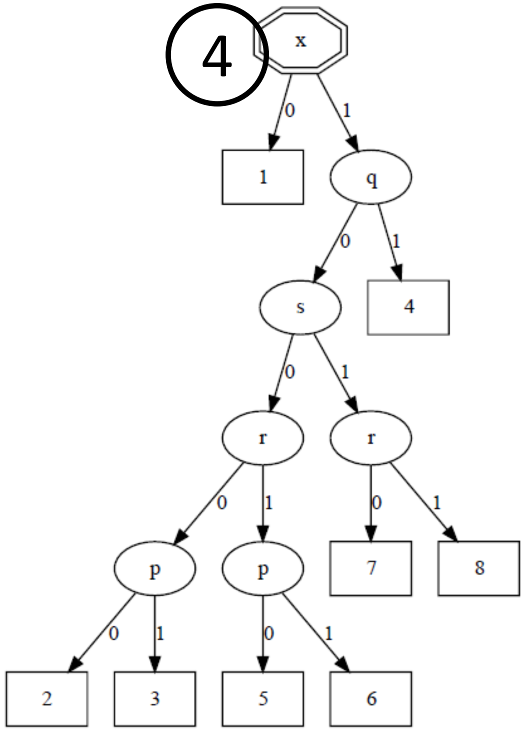
```
pixel_set:
  pixels:
    - {name: p, coords: [-1, -1]}
    - {name: x, coords: [ 0, 0]}
    - ...
  shifts: [1, 1]
conditions: [p, q, r, s, x]
actions: [nothing, x<-newlabel, x<-p, x<-q, x<-r, x
   <-p+r, x<-s, x<-r+s]
rules: [[1], [1], ..., [2], [3], [4], [3, 4], [5],
   ..., [3, 4, 5, 7]]
```

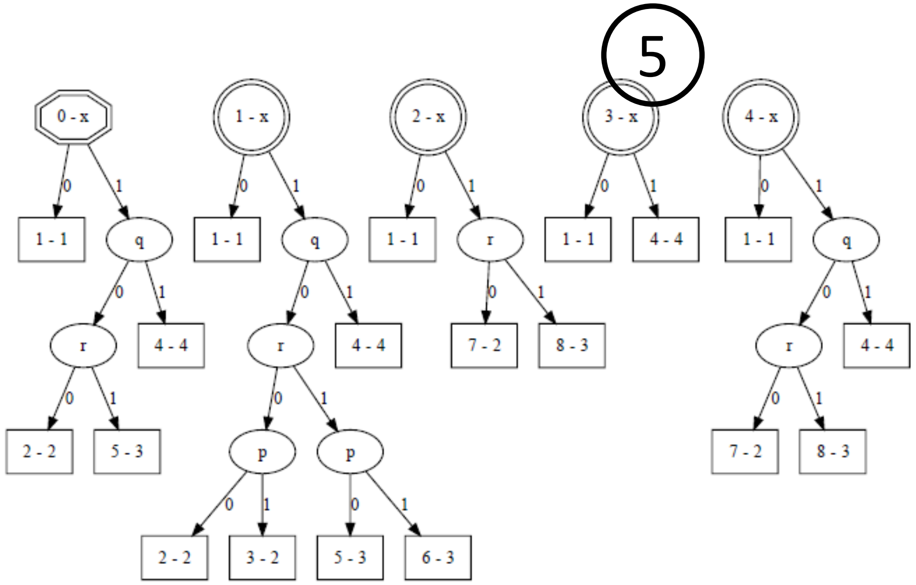
1

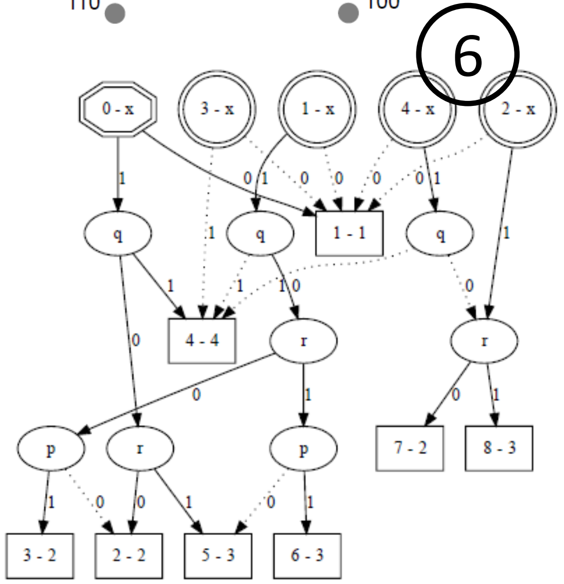
Conditions	x	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	condition outcomes
	p	-	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1	
	q	-	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1	
	r	-	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1	
	s	-	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1	
Actions	no op.	1																	action entries
	new		1																
	p			1				1		1				1	1			1	
	q				1			1			1	1		1	1		1	1	
	r					1					1			1			1	1	
	s						1			1		1			1		1	1	
	p + r								1							1			
	r + s												1			1			

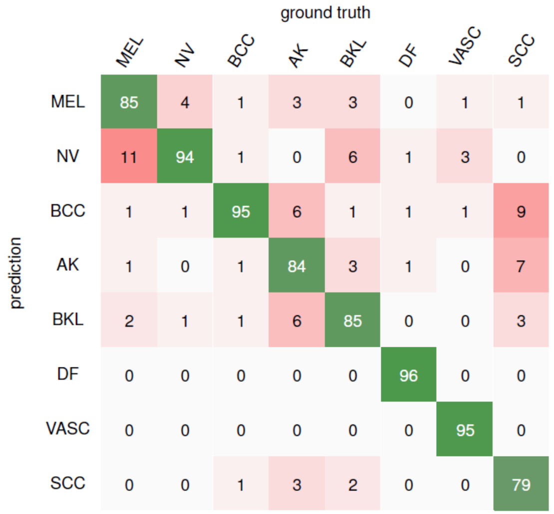
2



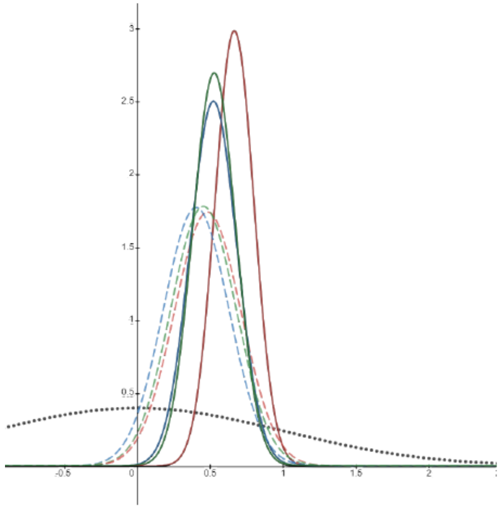


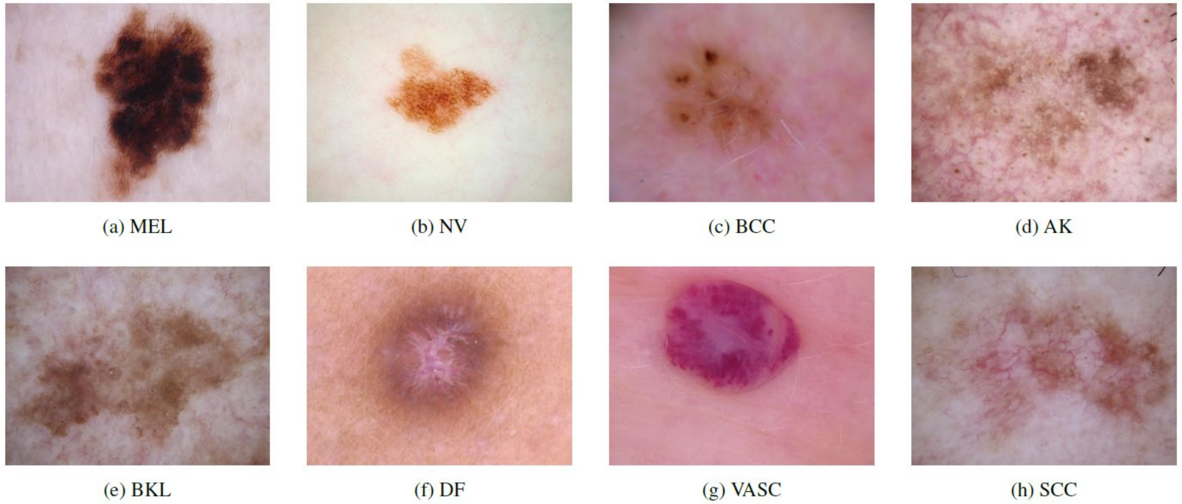


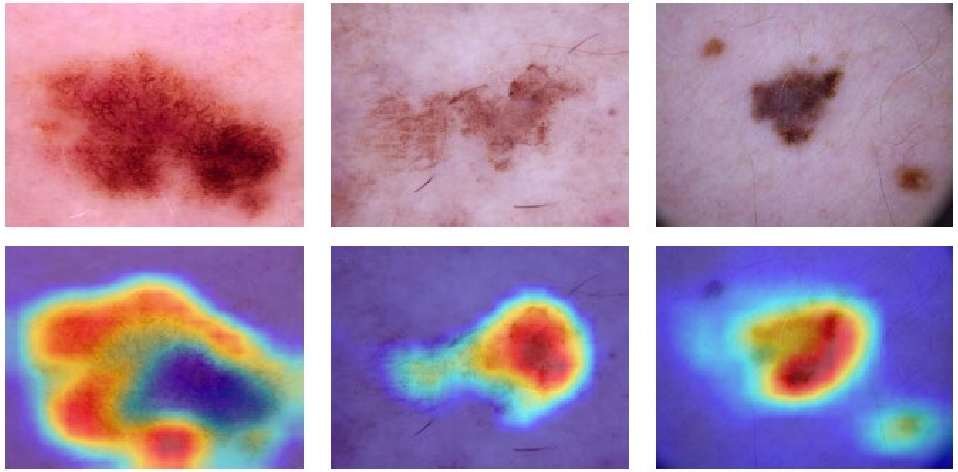


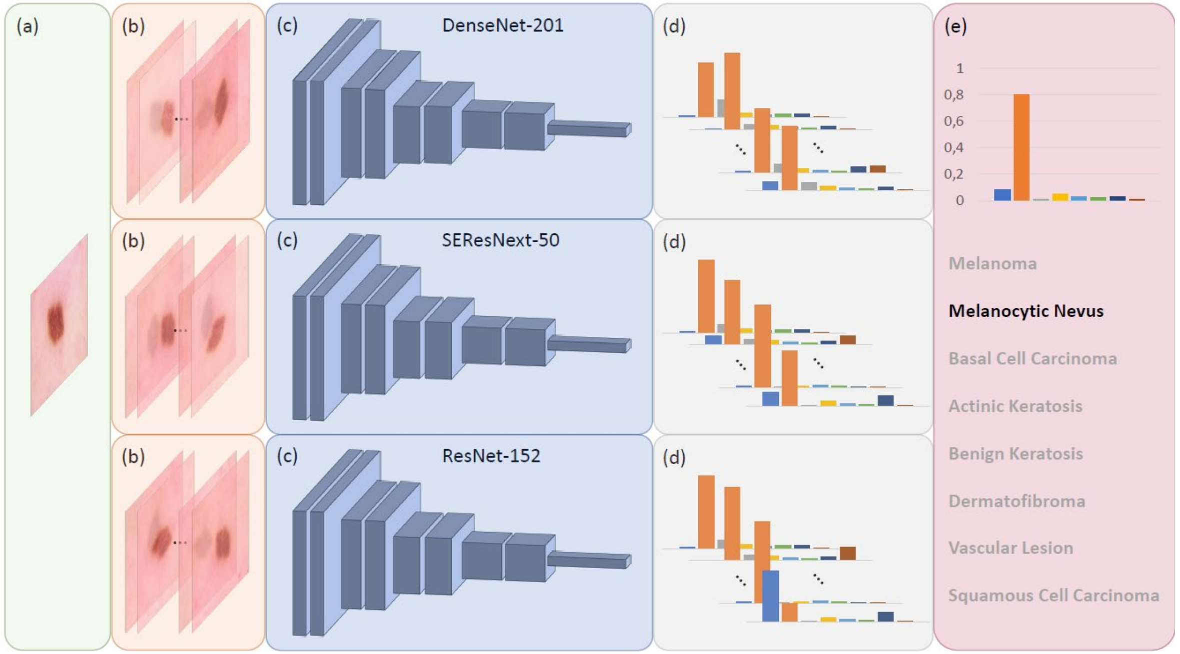


prediction \ ground truth	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
MEL	85	4	1	3	3	0	1	1
NV	11	94	1	0	6	1	3	0
BCC	1	1	95	6	1	1	1	9
AK	1	0	1	84	3	1	0	7
BKL	2	1	1	6	85	0	0	3
DF	0	0	0	0	0	96	0	0
VASC	0	0	0	0	0	0	95	0
SCC	0	0	1	3	2	0	0	79

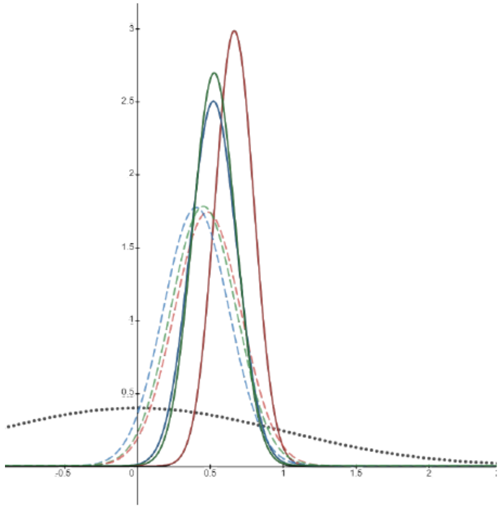


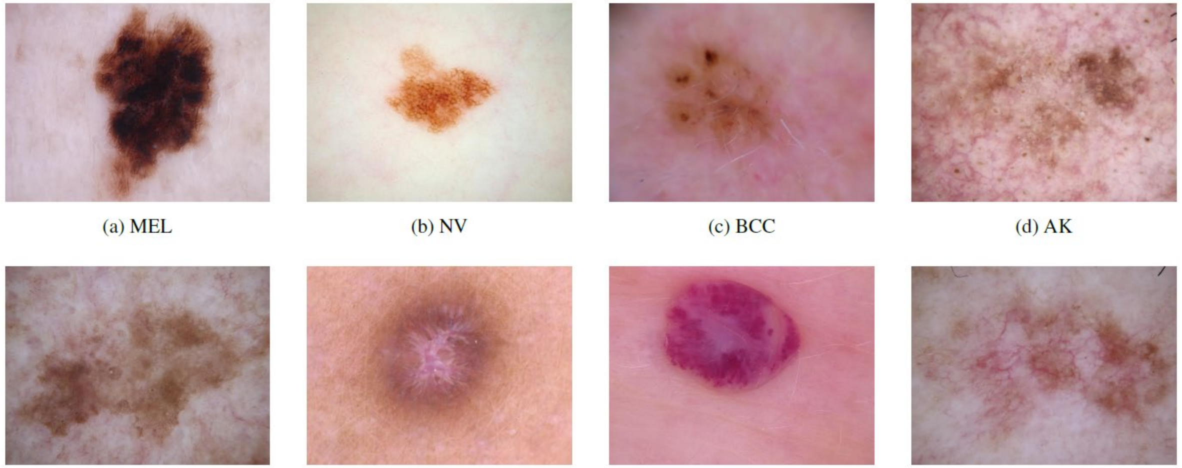






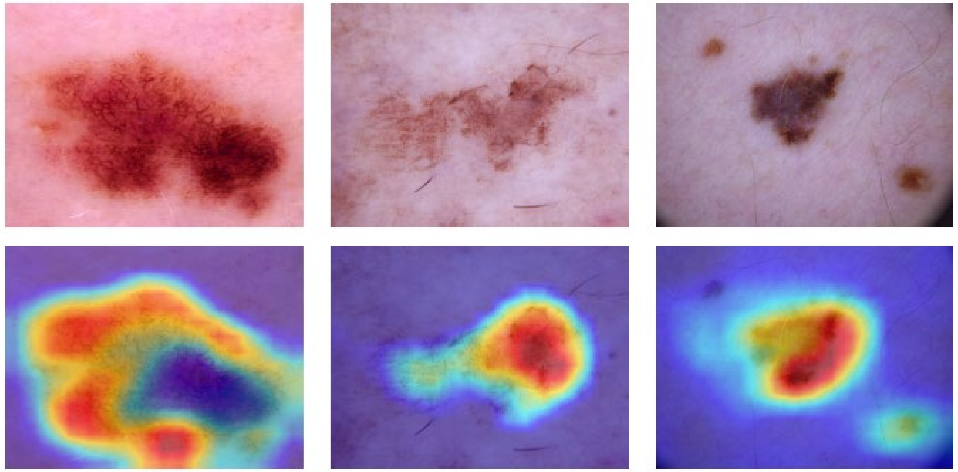
prediction \ ground truth	MEL	NV	BCC	AK	BKL	DF	VASC	SCC
MEL	85	4	1	3	3	0	1	1
NV	11	94	1	0	6	1	3	0
BCC	1	1	95	6	1	1	1	9
AK	1	0	1	84	3	1	0	7
BKL	2	1	1	6	85	0	0	3
DF	0	0	0	0	0	96	0	0
VASC	0	0	0	0	0	0	95	0
SCC	0	0	1	3	2	0	0	79

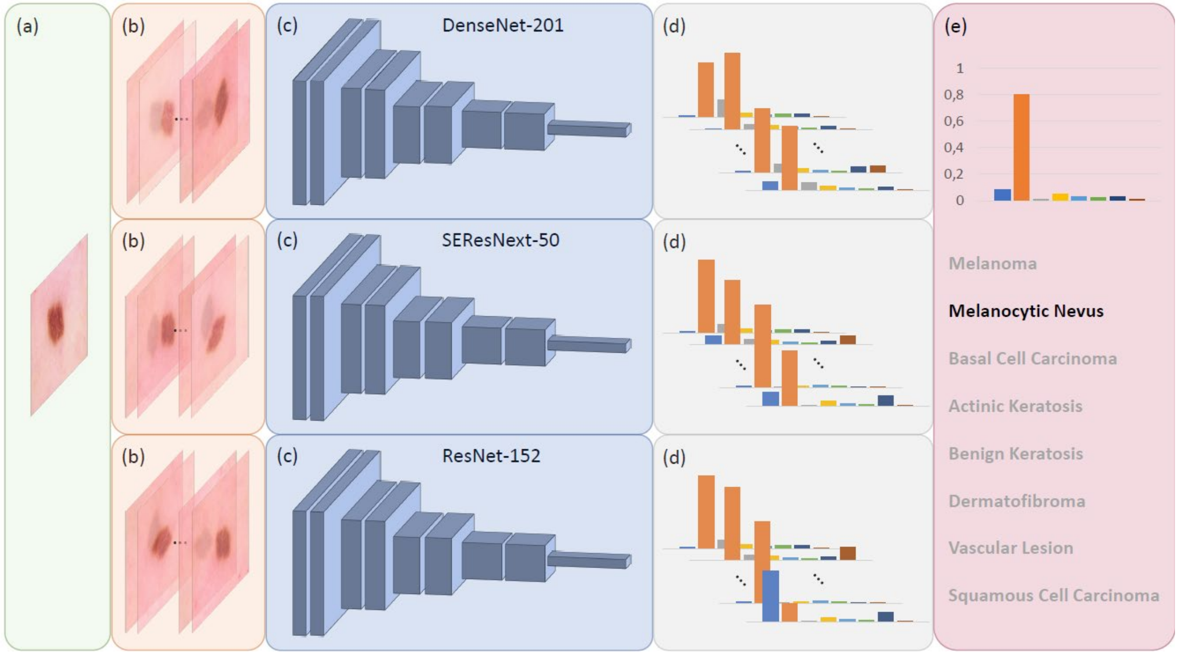


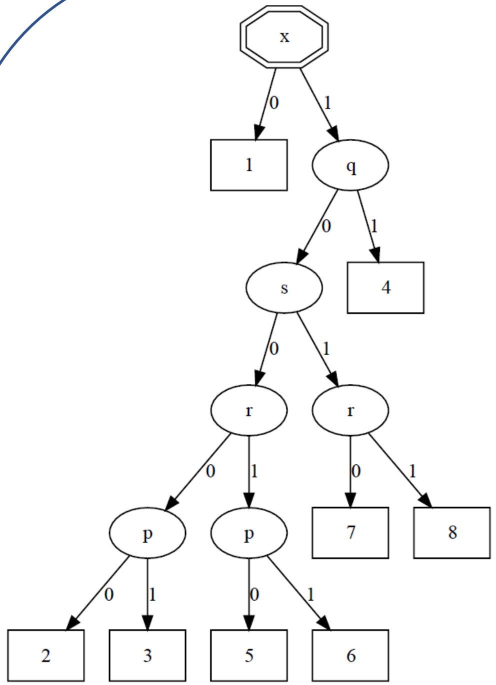


(a) MEL (b) NV (c) BCC (d) AK

(e) BKL (f) DF (g) VASC (h) SCC

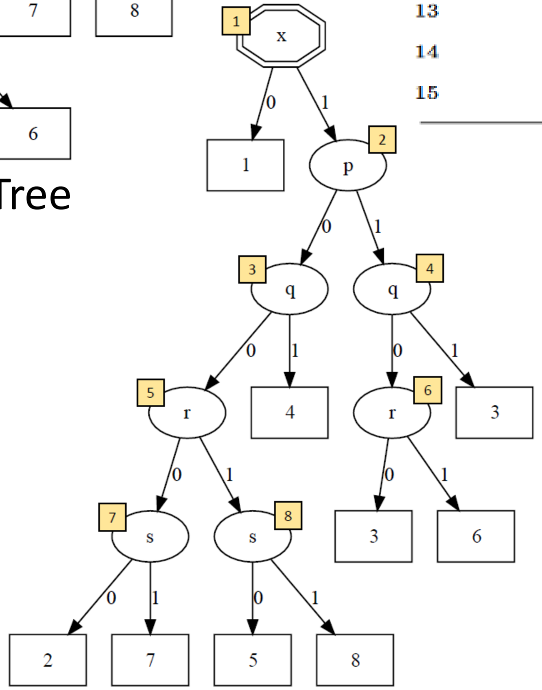






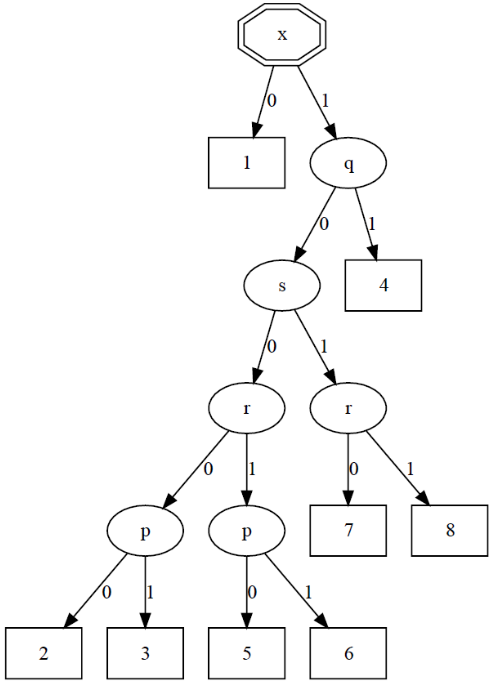
Optimal DTree

```yaml
CPU 3D 26-way connectivity:
  execute: true
  perform:
      correctness: true
      average: true
      average_with_steps: true
      density: false
      granularity: false
      memory: true
  algorithms:
      - EPDT_3D_19c_RemSP
      - EPDT_3D_22c_RemSP
      - EPDT_3D_26c_RemSP
      - LEB_3D_TTA
      - RBTS_3D_TTA
```



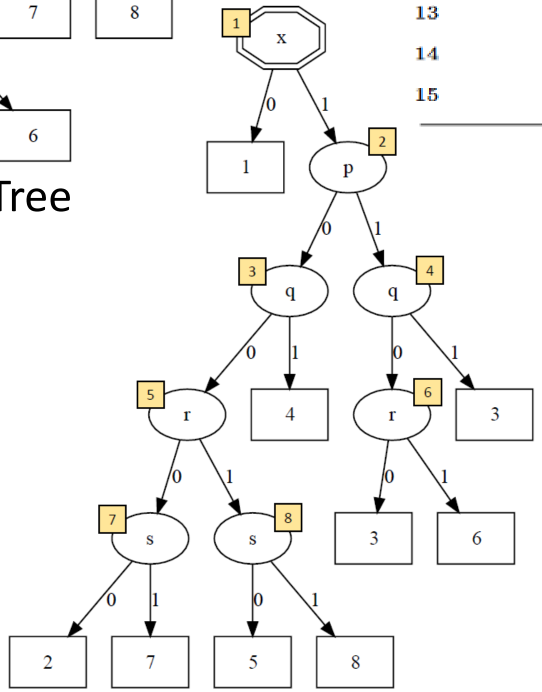
EPDT

Reproducible Label



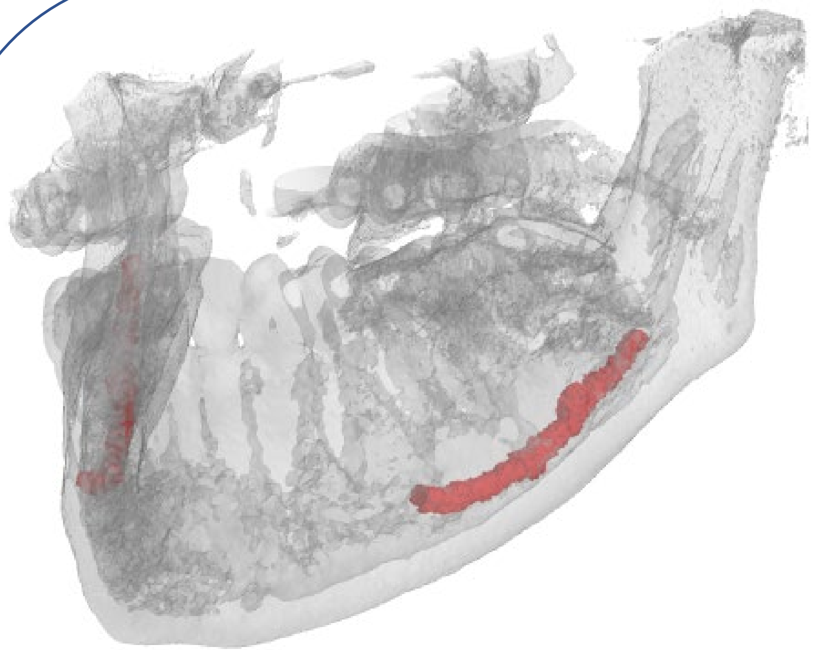
Optimal DTree

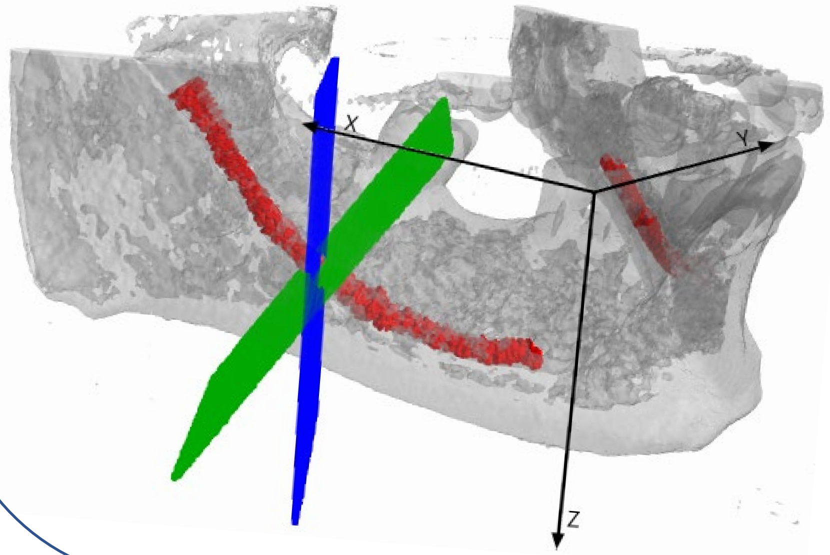
```yaml
CPU 3D 26-way connectivity:
  execute: true
  perform:
      correctness: true
      average: true
      average_with_steps: true
      density: false
      granularity: false
      memory: true
  algorithms:
      - EPDT_3D_19c_RemSP
      - EPDT_3D_22c_RemSP
      - EPDT_3D_26c_RemSP
      - LEB_3D_TTA
      - RBTS_3D_TTA
```



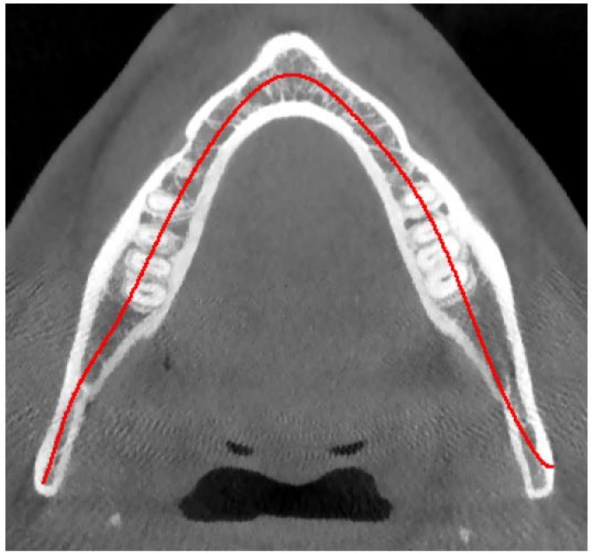
EPDT

Reproducible Label

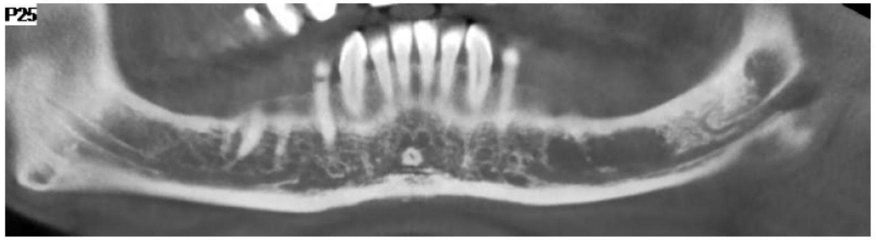




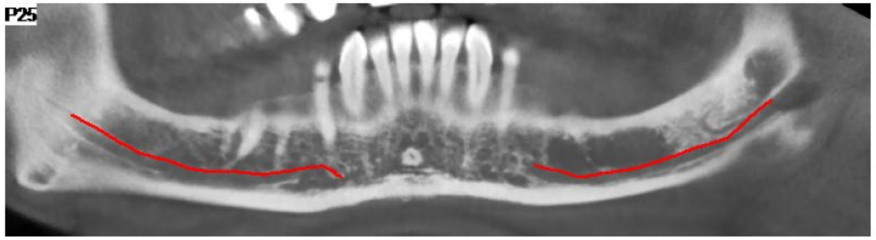
(a) Closing & Thresholding (b) CCL (c) Skeletonization (d) Polynomial Approx.



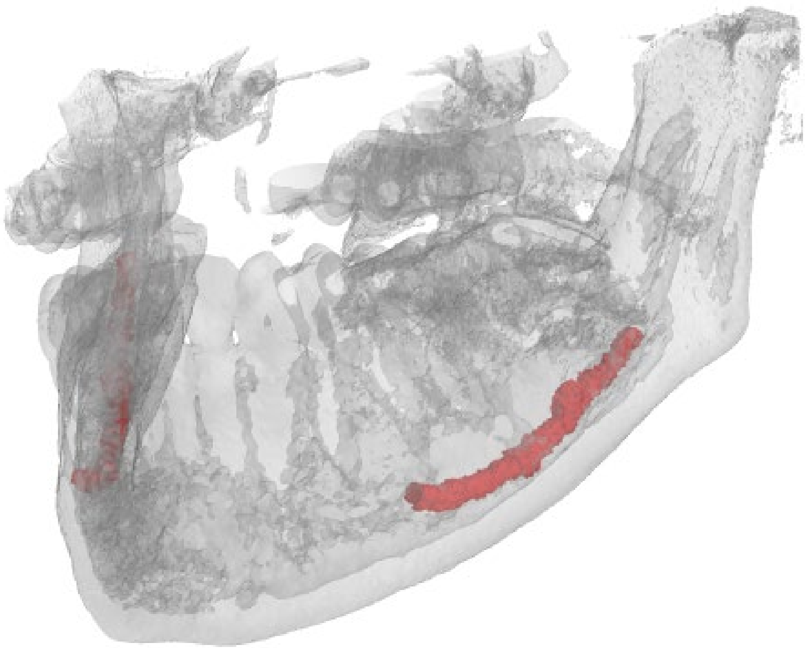
(a) Axial Slice

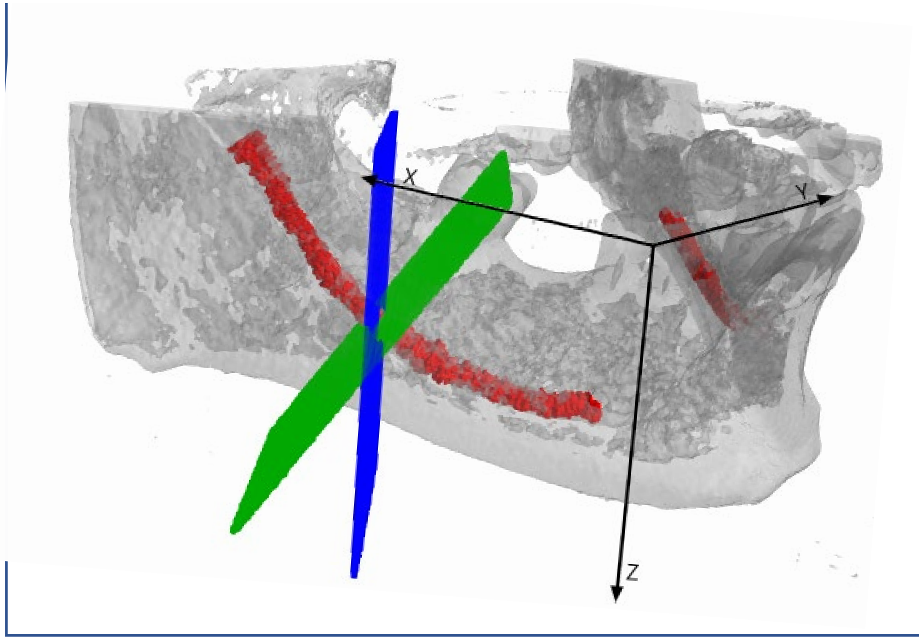


(b) Original Panoramic View

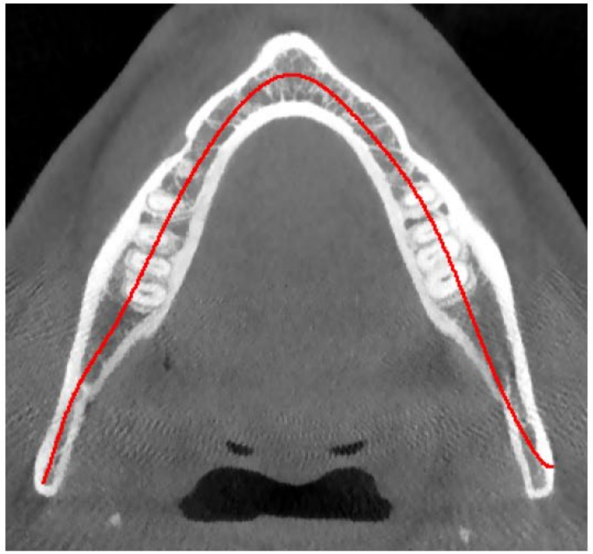


(c) Annotated Panoramic View

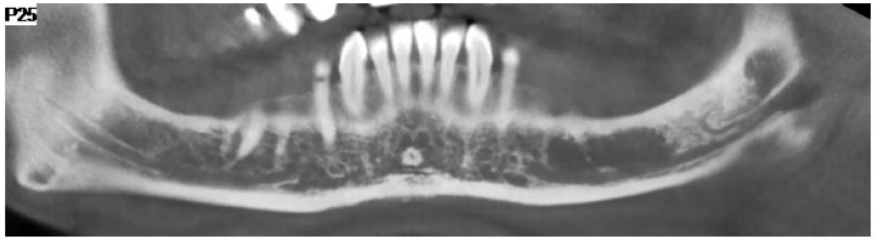




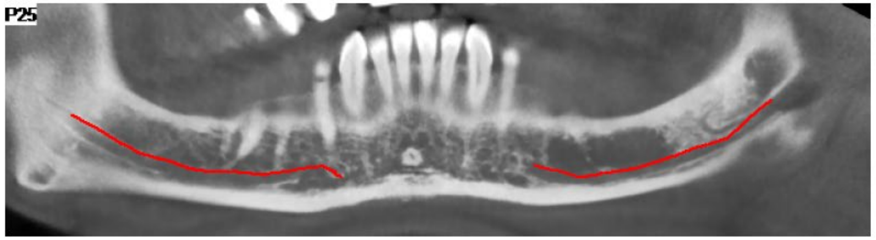
(a) Closing & Thresholding (b) CCL (c) Skeletonization (d) Polynomial Approx.



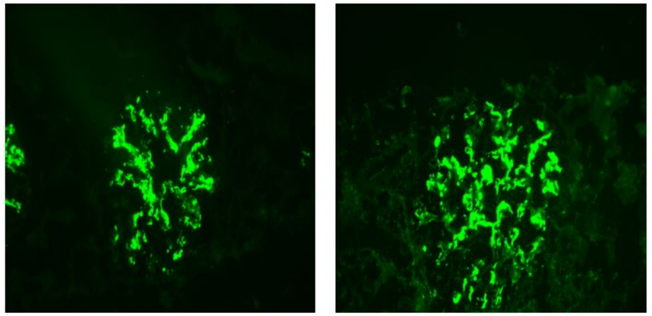
(a) Axial Slice



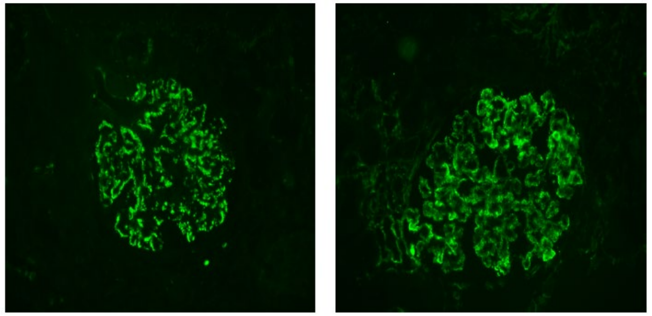
(b) Original Panoramic View



(c) Annotated Panoramic View



(a) Mesangial



(b) Parietal

AGREEMENT BETWEEN HUMAN EVALUATORS (THREE DIFFERENT EXPERT PATHOLOGISTS P1, P2, AND P3) AND GROUND TRUTH GT CALCULATED FOR BOTH MESANGIAL (A) AND PARIETAL (B) PATTERNS USING THE COHEN'S KAPPA.

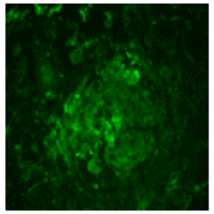
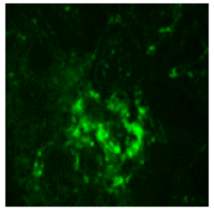
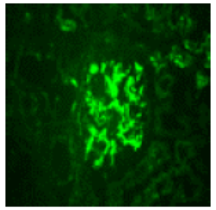
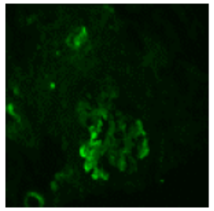
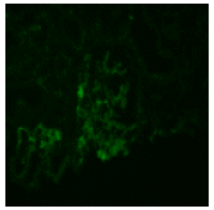
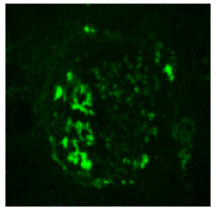
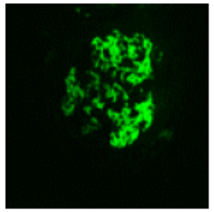
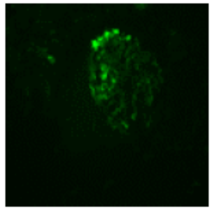
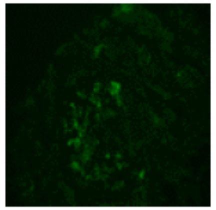
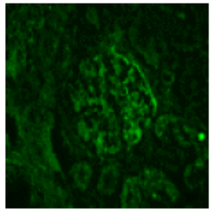
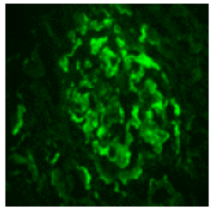
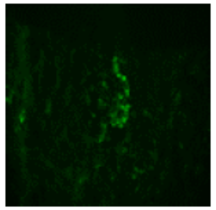
	GT	P1	P2
P3	0.50	0.70	0.34
P2	0.50	0.50	
P1	0.80		

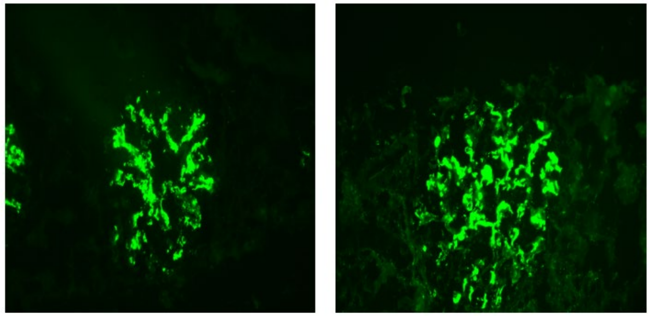
(a) Mesangial

	GT	P1	P2
P3	0.40	0.60	0.60
P2	0.40	0.42	
P1	0.60		

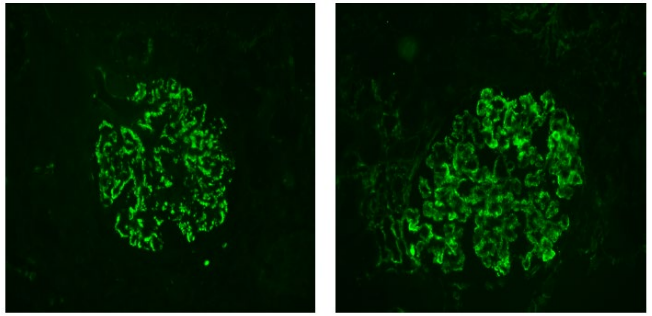
(b) Parietal

VISUALIZATION OF TEMPERATURE SCALING EFFECTIVENESS OVER DENSENET-121 WITH NO DROPOUT FOR THE MESANGIAL PATTERN RECOGNITION TASK. EACH COLUMN OF IMAGES IS IDENTIFIED BY THE CNN PREDICTION WITH RESPECT TO THE GROUND TRUTH ANNOTATION.

GT: yes **Pred: no**	***Calib*** **Uncalib**	**GT: no** **Pred: yes**	***Calib*** **Uncalib**	**GT: yes** **Pred: yes**	***Calib*** **Uncalib** ***Human***
	*0.830* 0.992		*0.781* 0.980		*0.965* 0.999 *1.000*
	*0.771* 0.977		*0.774* 0.964		*0.771* 0.977 *0.400*
	*0.571* 0.707		*0.572* 0.711		*0.658* 0.883 *0.600*
	*0.562* 0.684		*0.560* 0.679		*0.558* 0.673 *0.400*



(a) Mesangial



(b) Parietal

AGREEMENT BETWEEN HUMAN EVALUATORS (THREE DIFFERENT EXPERT PATHOLOGISTS P1, P2, AND P3) AND GROUND TRUTH GT CALCULATED FOR BOTH MESANGIAL (A) AND PARIETAL (B) PATTERNS USING THE COHEN'S KAPPA.

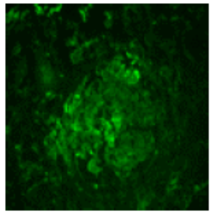
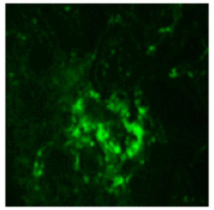
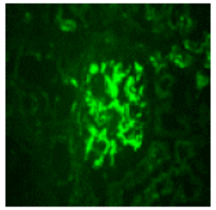
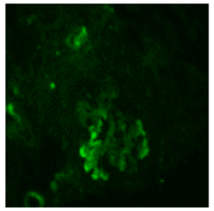
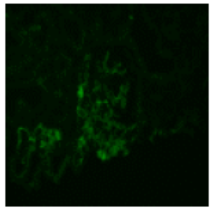
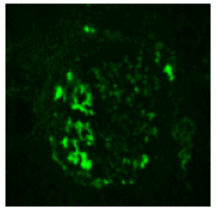
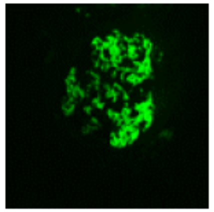
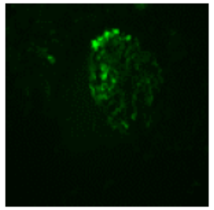
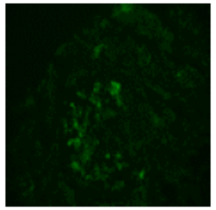
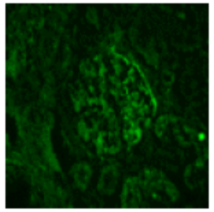
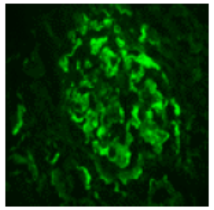
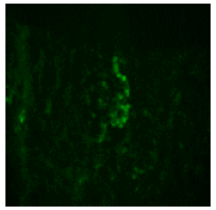
	GT	**P1**	**P2**
P3	0.50	0.70	0.34
P2	0.50	0.50	
P1	0.80		

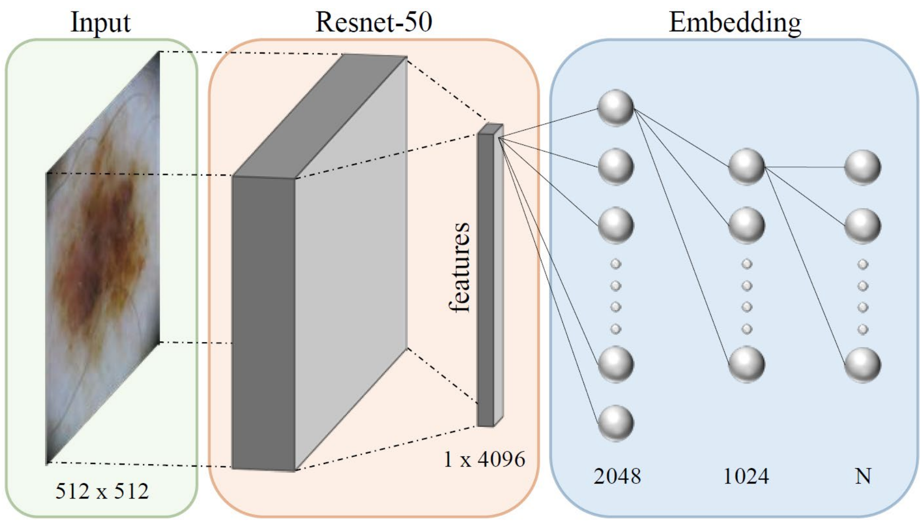
(a) Mesangial

	GT	**P1**	**P2**
P3	0.40	0.60	0.60
P2	0.40	0.42	
P1	0.60		

(b) Parietal

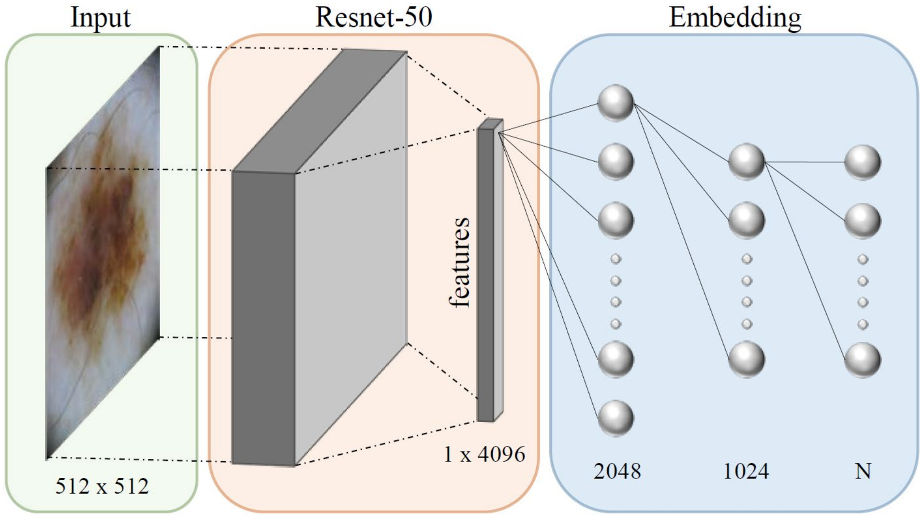
VISUALIZATION OF TEMPERATURE SCALING EFFECTIVENESS OVER DENSENET-121 WITH NO DROPOUT FOR THE MESANGIAL PATTERN RECOGNITION TASK. EACH COLUMN OF IMAGES IS IDENTIFIED BY THE CNN PREDICTION WITH RESPECT TO THE GROUND TRUTH ANNOTATION.

GT: yes **Pred: no**	***Calib*** **Uncalib**	**GT: no** **Pred: yes**	***Calib*** **Uncalib**	**GT: yes** **Pred: yes**	***Calib*** **Uncalib** ***Human***
	*0.830* 0.992		*0.781* 0.980		*0.965* 0.999 *1.000*
	*0.771* 0.977		*0.774* 0.964		*0.771* 0.977 *0.400*
	*0.571* 0.707		*0.572* 0.711		*0.658* 0.883 *0.600*
	*0.562* 0.684		*0.560* 0.679		*0.558* 0.673 *0.400*



AVERAGE PRECISION AT K MEASURED FOR EVERY MODEL ANALYZED, FOR THREE VALUES OF K, 1, 5, AND 10. CENTRAL COLUMNS REPORT AVERAGE VALUES SEPARATED FOR EACH CLASS, AND THE LAST COLUMN REPORTS THE BALANCED AVERAGE. NOVEL PROPOSALS ARE IDENTIFIED BY *.

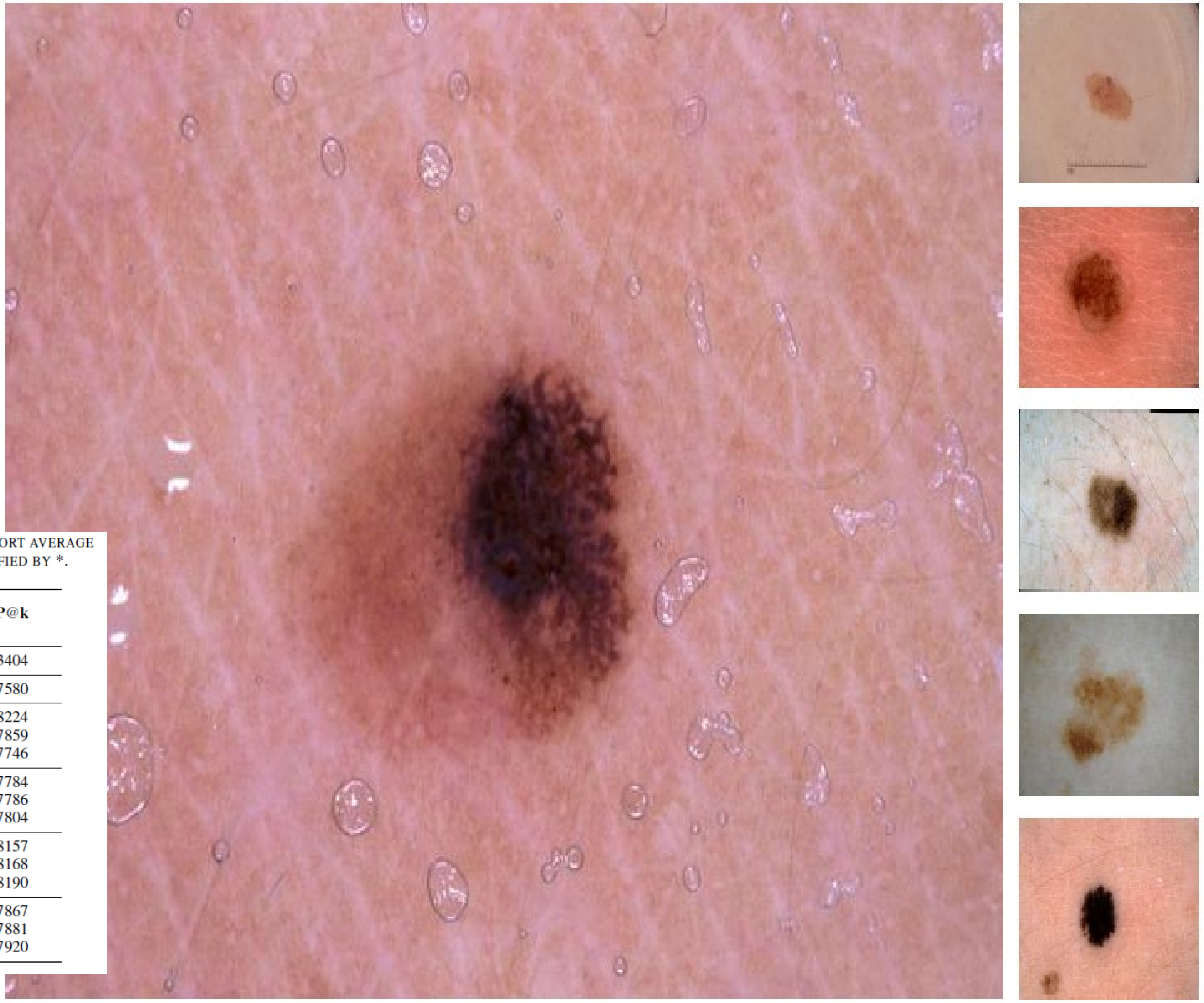
Model	Cut-Off k	Per class P@k								AP@k
		MEL	NV	BCC	AK	BKL	DF	VASC	SCC	
Hash-AP [45]	-	0.4786	0.6111	0.5730	0.1896	0.1984	0.1505	0.3842	0.1375	0.3404
Hash-AP ResNet*	-	0.8176	0.7558	0.8509	0.7417	0.6256	0.7604	0.8271	0.6851	0.7580
Classification*	1	0.7840	0.9369	0.9347	0.7400	0.8300	0.7733	0.8667	0.7133	0.8224
	5	0.7262	0.9111	0.9029	0.7190	0.7724	0.7333	0.8373	0.6853	0.7859
	10	0.7040	0.9038	0.8957	0.7160	0.7470	0.7213	0.8307	0.6787	0.7746
Embedding End-to-End*	1	0.7400	0.9018	0.9133	0.6600	0.7520	0.7333	0.8267	0.7000	0.7784
	5	0.7314	0.8923	0.9005	0.6820	0.7576	0.7387	0.8240	0.7027	0.7786
	10	0.7322	0.8905	0.8993	0.6855	0.7572	0.7440	0.8253	0.7093	0.7804
Class & Embedding*	1	0.7490	0.9347	0.8973	0.7150	0.7700	0.8400	0.9067	0.7133	0.8157
	5	0.7542	0.9347	0.9013	0.7170	0.7768	0.8400	0.9013	0.7093	0.8168
	10	0.7531	0.9331	0.9032	0.7170	0.7814	0.8453	0.9040	0.7147	0.8190
Class & Embedding End-to-End*	1	0.7560	0.9022	0.9027	0.6600	0.7600	0.7733	0.8267	0.7133	0.7867
	5	0.7458	0.9030	0.8992	0.6770	0.7588	0.7707	0.8293	0.7213	0.7881
	10	0.7436	0.9012	0.9009	0.6830	0.7668	0.7760	0.8373	0.7273	0.7920

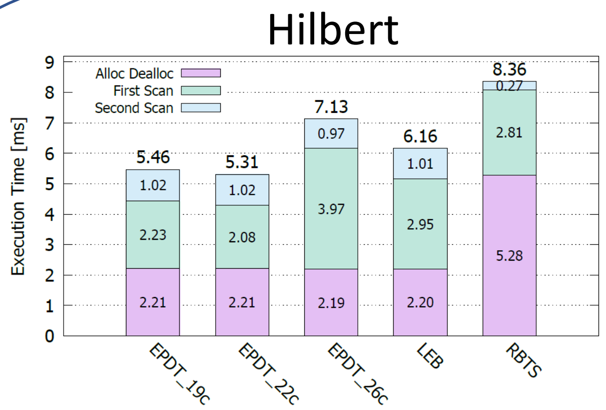


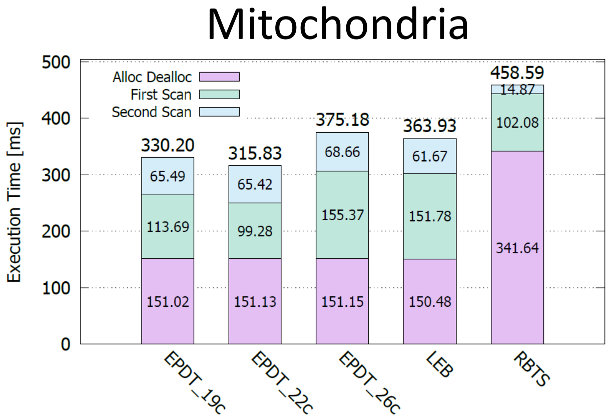
AVERAGE PRECISION AT K MEASURED FOR EVERY MODEL ANALYZED, FOR THREE VALUES OF K, 1, 5, AND 10. CENTRAL COLUMNS REPORT AVERAGE VALUES SEPARATED FOR EACH CLASS, AND THE LAST COLUMN REPORTS THE BALANCED AVERAGE. NOVEL PROPOSALS ARE IDENTIFIED BY *.

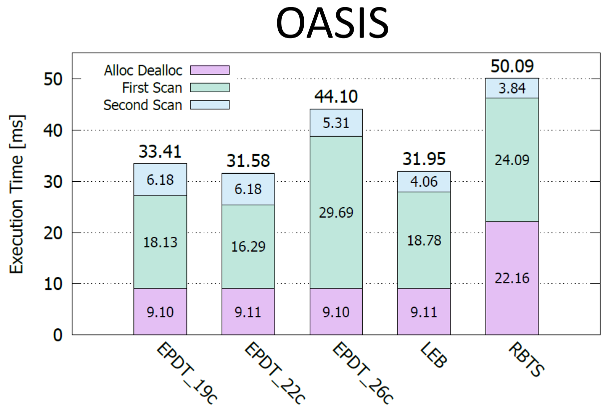
Model	Cut-Off k	Per class P@k								AP@k
		MEL	NV	BCC	AK	BKL	DF	VASC	SCC	
Hash-AP [45]	-	0.4786	0.6111	0.5730	0.1896	0.1984	0.1505	0.3842	0.1375	0.3404
Hash-AP ResNet*	-	0.8176	0.7558	0.8509	0.7417	0.6256	0.7604	0.8271	0.6851	0.7580
Classification*	1	0.7840	0.9369	0.9347	0.7400	0.8300	0.7733	0.8667	0.7133	0.8224
	5	0.7262	0.9111	0.9029	0.7190	0.7724	0.7333	0.8373	0.6853	0.7859
	10	0.7040	0.9038	0.8957	0.7160	0.7470	0.7213	0.8307	0.6787	0.7746
Embedding End-to-End*	1	0.7400	0.9018	0.9133	0.6600	0.7520	0.7333	0.8267	0.7000	0.7784
	5	0.7314	0.8923	0.9005	0.6820	0.7576	0.7387	0.8240	0.7027	0.7786
	10	0.7322	0.8905	0.8993	0.6855	0.7572	0.7440	0.8253	0.7093	0.7804
Class & Embedding*	1	0.7490	0.9347	0.8973	0.7150	0.7700	0.8400	0.9067	0.7133	0.8157
	5	0.7542	0.9347	0.9013	0.7170	0.7768	0.8400	0.9013	0.7093	0.8168
	10	0.7531	0.9331	0.9032	0.7170	0.7814	0.8453	0.9040	0.7147	0.8190
Class & Embedding End-to-End*	1	0.7560	0.9022	0.9027	0.6600	0.7600	0.7733	0.8267	0.7133	0.7867
	5	0.7458	0.9030	0.8992	0.6770	0.7588	0.7707	0.8293	0.7213	0.7881
	10	0.7436	0.9012	0.9009	0.6830	0.7668	0.7760	0.8373	0.7273	0.7920

Ranking System



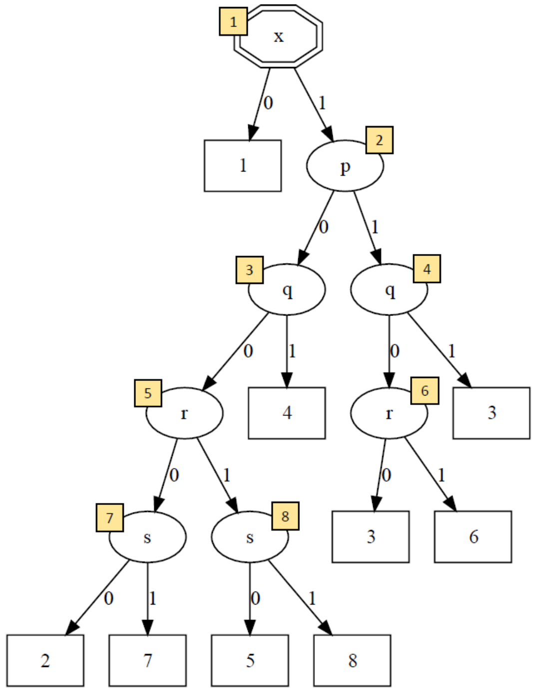




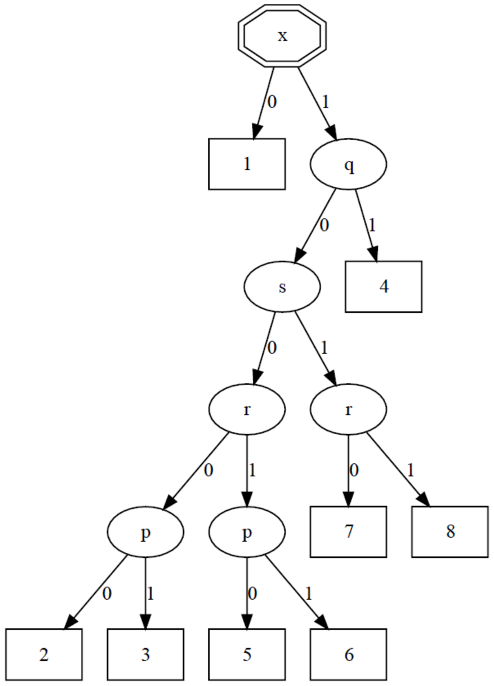


AVERAGE RESULTS IN MS. LOWER IS BETTER.

	Hilbert	Mitochondria	OASIS
EPDT_19c_RemSP	4.77	276.72	30.12
EPDT_19c_TTA	5.07	281.07	30.54
EPDT_19c_UF	4.78	277.66	30.38
EPDT_19c_UFPC	4.81	277.63	30.57
EPDT_22c_RemSP	**4.68**	276.48	29.07
EPDT_22c_TTA	4.84	276.66	**29.00**
EPDT_22c_UF	4.73	276.79	29.16
EPDT_22c_UFPC	4.69	**271.62**	29.36
EPDT_26c_RemSP	7.96	398.39	51.22
EPDT_26c_TTA	8.15	398.30	51.45
EPDT_26c_UF	8.02	400.45	51.61
EPDT_26c_UFPC	7.93	399.57	51.62
LEB_TTA	5.84	313.63	29.03
RBTS_TTA	8.24	430.46	45.50



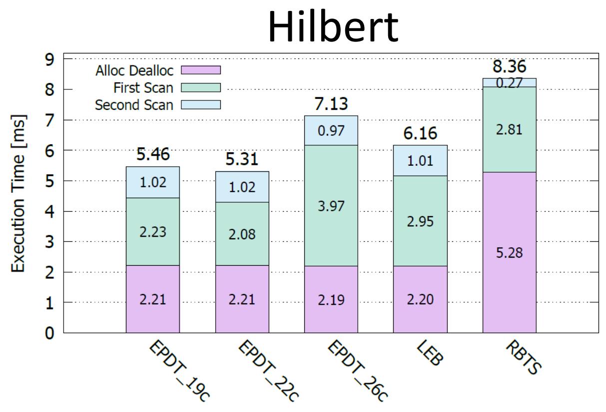
EPDT

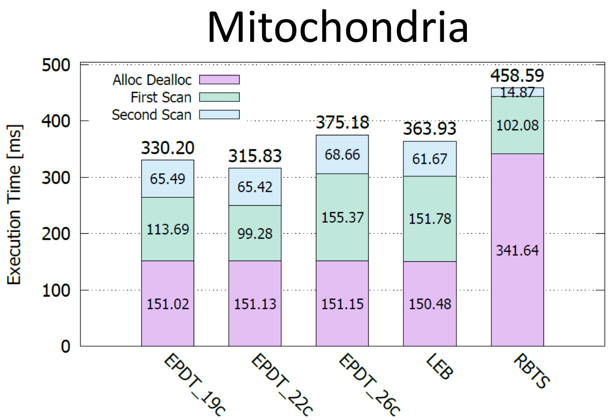


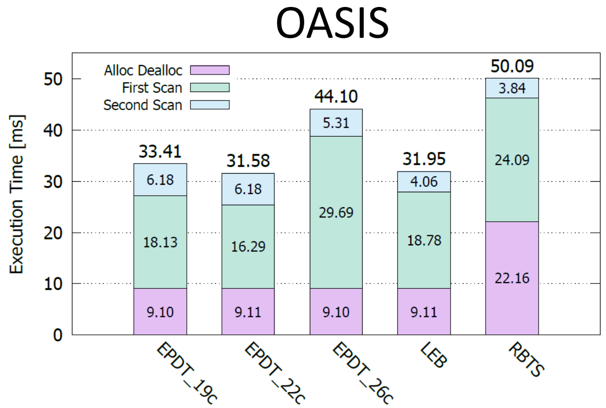
Optimal DTree

ENTROPY AND INFORMATION GAIN FOR THE GENERATION OF THE SAUF TREE USING THE TEMPORAL POPULARITY CLASSIFIER. EACH ROW CORRESPONDS TO ONE NODE OF THE TREE. VALUES HAVE BEEN ROUNDED FOR VISUALIZATION REASONS. BOLD VALUES IDENTIFY THE ENTROPY AND THE IG OF THE CONDITION CHOSEN FOR THE CURRENT NODE IN THE DECISION TREE. THE FINAL RESULTING TREE IS REPORTED ABOVE TO THE LEFT.

Node	Depth	$H(S)$	p			q			r			s			x		
			$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG
1	0	2.2	2.0	1.4	0.5	2.3	1.5	0.3	1.9	2.1	0.2	2.1	2.1	0.1	**0.0**	**2.4**	**1.0**
2	1	2.4	**2.0**	**0.8**	**1.0**	2.5	1.0	0.7	1.8	2.3	0.4	2.2	2.2	0.2			
3	2	2.0				**2.0**	**0.0**	**1.0**	1.5	1.5	0.5	1.5	1.5	0.5			
4	2	0.8				**1.0**	**0.0**	**0.3**	0.0	1.0	0.3	0.8	0.8	0.0			
5	3	2.0							**1.0**	**1.0**	**1.0**	1.0	1.0	1.0			
6	3	1.0							**0.0**	**0.0**	**1.0**	1.0	1.0	0.0			
7	4	1.0										**0.0**	**0.0**	**1.0**			
8	4	1.0										**0.0**	**0.0**	**1.0**			

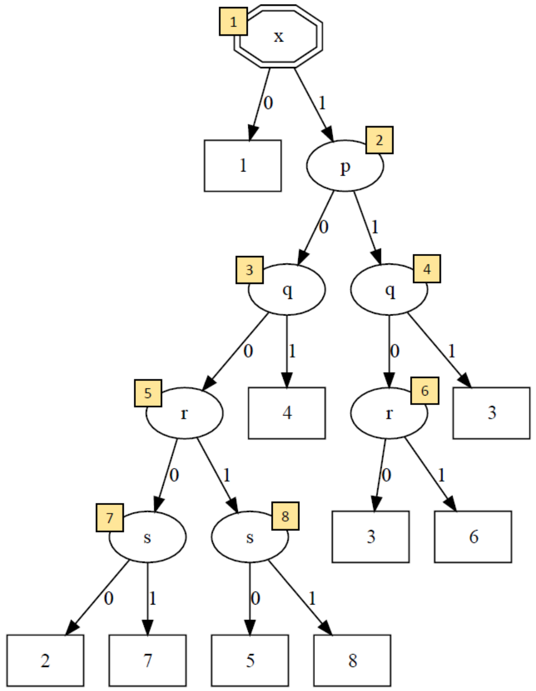




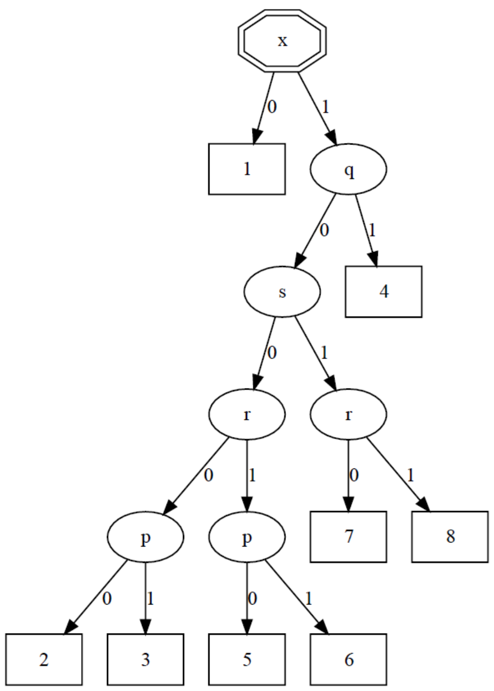


Average Results in ms. Lower is better.

	Hilbert	Mitochondria	OASIS
EPDT_19c_RemSP	4.77	276.72	30.12
EPDT_19c_TTA	5.07	281.07	30.54
EPDT_19c_UF	4.78	277.66	30.38
EPDT_19c_UFPC	4.81	277.63	30.57
EPDT_22c_RemSP	**4.68**	276.48	29.07
EPDT_22c_TTA	4.84	276.66	**29.00**
EPDT_22c_UF	4.73	276.79	29.16
EPDT_22c_UFPC	4.69	**271.62**	29.36
EPDT_26c_RemSP	7.96	398.39	51.22
EPDT_26c_TTA	8.15	398.30	51.45
EPDT_26c_UF	8.02	400.45	51.61
EPDT_26c_UFPC	7.93	399.57	51.62
LEB_TTA	5.84	313.63	29.03
RBTS_TTA	8.24	430.46	45.50



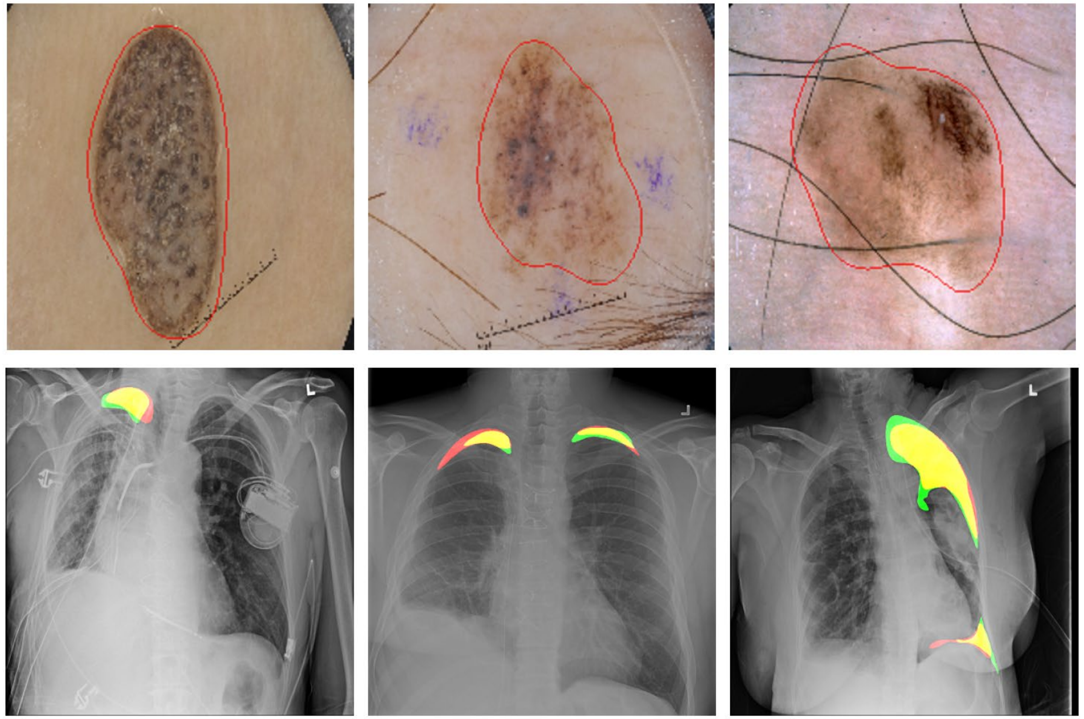
EPDT

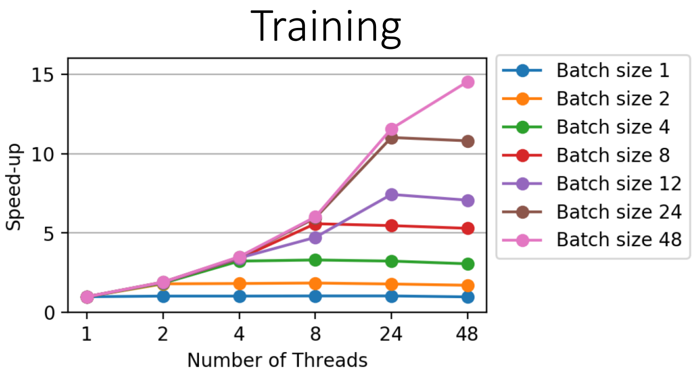


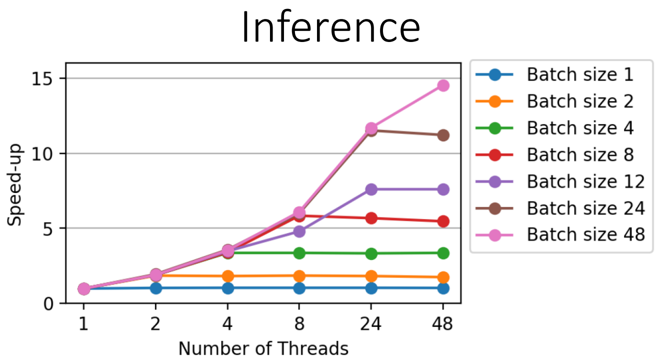
Optimal DTree

Entropy and information gain for the generation of the SAUF tree using the temporal popularity classifier. Each row corresponds to one node of the tree. Values have been rounded for visualization reasons. Bold values identify the entropy and the IG of the condition chosen for the current node in the decision tree. The final resulting tree is reported above to the left.

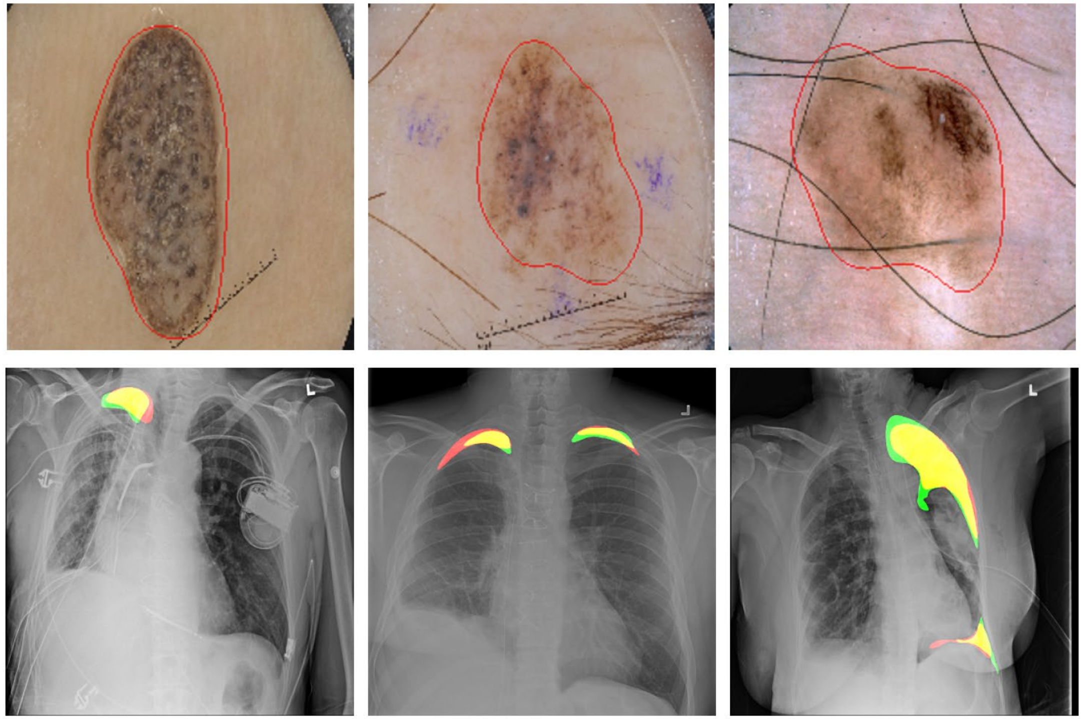
Node	Depth	$H(S)$	p			q			r			s			x		
			$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG	$H(T_0)$	$H(T_1)$	IG
1	0	2.2	2.0	1.4	0.5	2.3	1.5	0.3	1.9	2.1	0.2	2.1	2.1	0.1	**0.0**	**2.4**	**1.0**
2	1	2.4	**2.0**	**0.8**	**1.0**	2.5	1.0	0.7	1.8	2.3	0.4	2.2	2.2	0.2			
3	2	2.0				**2.0**	**0.0**	**1.0**	1.5	1.5	0.5	1.5	1.5	0.5			
4	2	0.8				**1.0**	**0.0**	**0.3**	0.0	1.0	0.3	0.8	0.8	0.0			
5	3	2.0							**1.0**	**1.0**	**1.0**	1.0	1.0	1.0			
6	3	1.0							**0.0**	**0.0**	**1.0**	1.0	1.0	0.0			
7	4	1.0										**0.0**	**0.0**	**1.0**			
8	4	1.0										**0.0**	**0.0**	**1.0**			

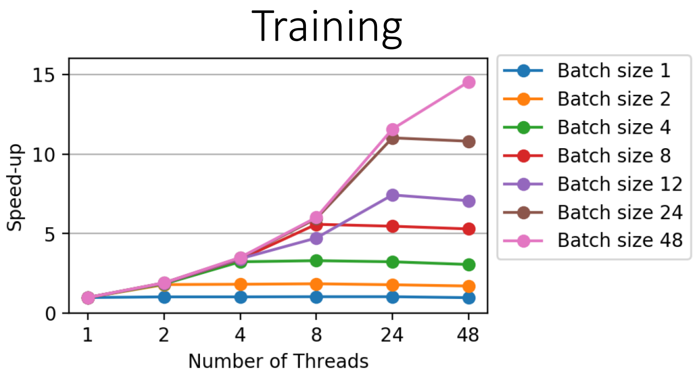


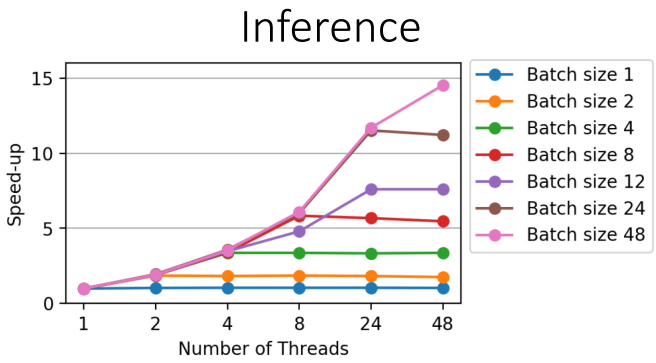




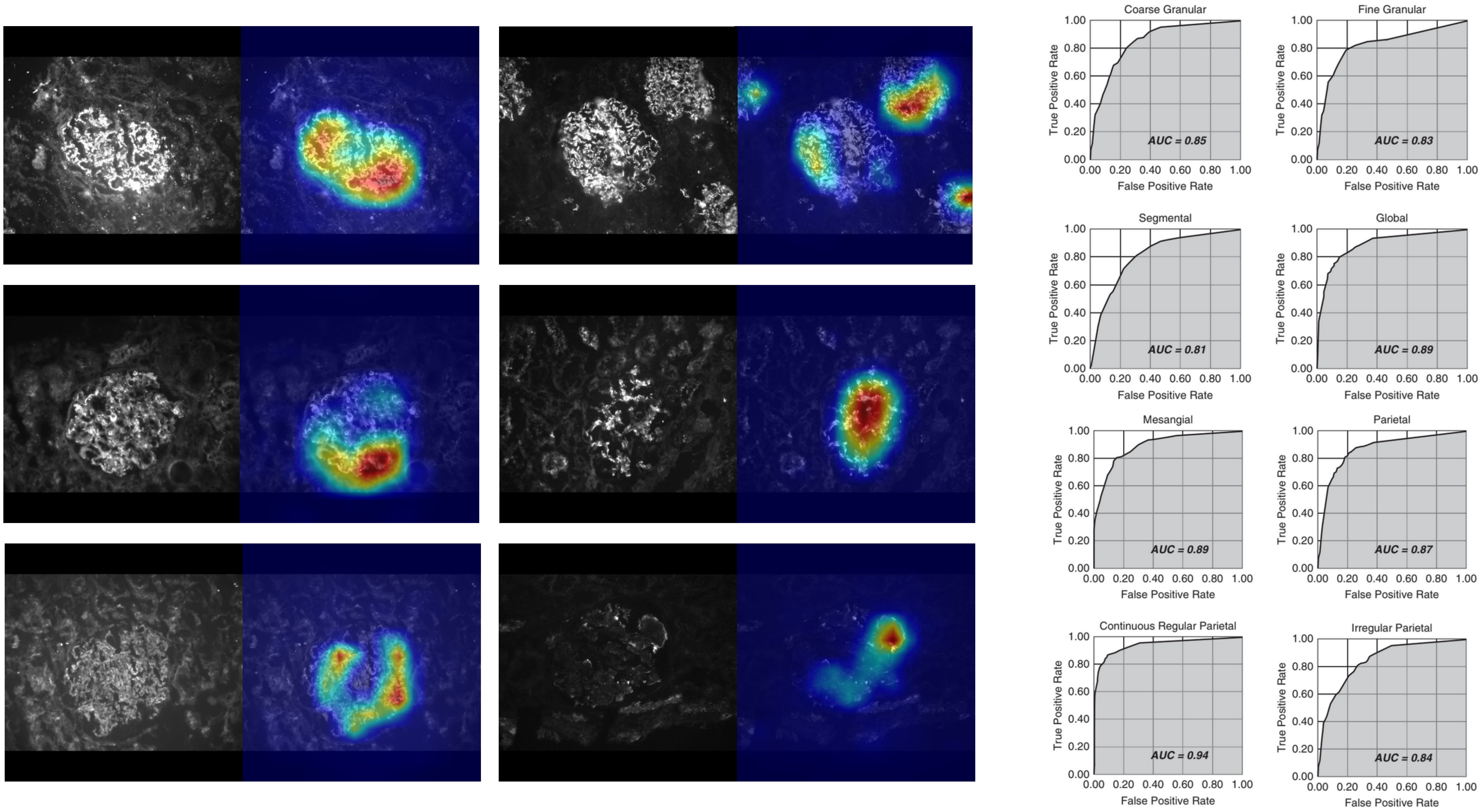

```python
import pyeddl.eddl as eddl
import pyeddl.eddlT as eddlT

epochs, batch_size, nclass = 10, 100, 10

# Build model
in_ = eddl.Input([784])
layer = in_
layer = eddl.ReLu(eddl.Dense(layer, 1024))
layer = eddl.ReLu(eddl.Dense(layer, 1024))
layer = eddl.ReLu(eddl.Dense(layer, 1024))
out = eddl.Softmax(eddl.Dense(layer, nclass))

net = eddl.Model([in_], [out])
eddl.build(net,
    eddl.rmsprop(0.01),               # Optimizer
    ["soft_cross_entropy"],           # Loss
    ["categorical_accuracy"],         # Metric
    eddl.CS_GPU([1], mem="low_mem")) # One GPU

# Load training and test data
x_tr = eddlT.load("trX.bin")
y_tr = eddlT.load("trY.bin")
x_ts = eddlT.load("tsX.bin")
y_ts = eddlT.load("tsY.bin")

# Preprocessing
eddlT.div_(x_tr, 255.0)
eddlT.div_(x_ts, 255.0)

# Train model
eddl.fit(net, [x_tr], [y_tr], batch_size, epochs)

# Evaluate
eddl.evaluate(net, [x_ts], [y_ts])
```



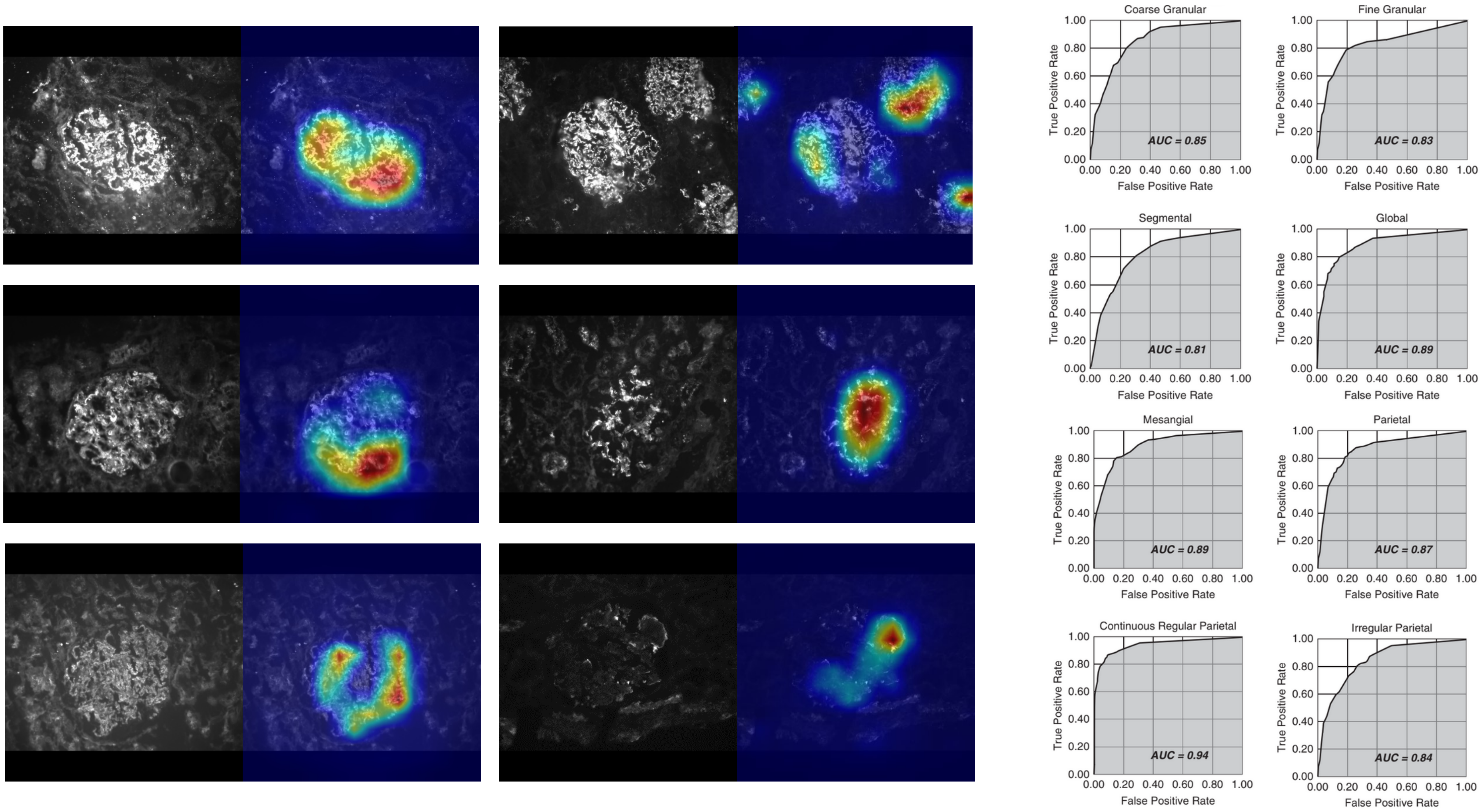





```python
import pyeddl.eddl as eddl
import pyeddl.eddlT as eddlT

epochs, batch_size, nclass = 10, 100, 10

# Build model
in_ = eddl.Input([784])
layer = in_
layer = eddl.ReLu(eddl.Dense(layer, 1024))
layer = eddl.ReLu(eddl.Dense(layer, 1024))
layer = eddl.ReLu(eddl.Dense(layer, 1024))
out = eddl.Softmax(eddl.Dense(layer, nclass))

net = eddl.Model([in_], [out])
eddl.build(net,
    eddl.rmsprop(0.01),                    # Optimizer
    ["soft_cross_entropy"],                # Loss
    ["categorical_accuracy"],              # Metric
    eddl.CS_GPU([1], mem="low_mem"))  # One GPU

# Load training and test data
x_tr = eddlT.load("trX.bin")
y_tr = eddlT.load("trY.bin")
x_ts = eddlT.load("tsX.bin")
y_ts = eddlT.load("tsY.bin")

# Preprocessing
eddlT.div_(x_tr, 255.0)
eddlT.div_(x_ts, 255.0)

# Train model
eddl.fit(net, [x_tr], [y_tr], batch_size, epochs)

# Evaluate
eddl.evaluate(net, [x_ts], [y_ts])
```



Coarse Granular
AUC = 0.85
Fine Granular
AUC = 0.83
Segmental
AUC = 0.81
Global
AUC = 0.89
Mesangial
AUC = 0.89
Parietal
AUC = 0.87
Continuous Regular Parietal
AUC = 0.94
Irregular Parietal
AUC = 0.84
True Positive Rate
False Positive Rate
0.00
0.20
0.40
0.60
0.80
1.00



Coarse Granular
AUC = 0.85
Fine Granular
AUC = 0.83
Segmental
AUC = 0.81
Global
AUC = 0.89
Mesangial
AUC = 0.89
Parietal
AUC = 0.87
Continuous Regular Parietal
AUC = 0.94
Irregular Parietal
AUC = 0.84
True Positive Rate
False Positive Rate
0.00
0.20
0.40
0.60
0.80
1.00

Algorithm 1: This algorithm provides the Cederberg RC-code action implementation, *i.e.* the instructions required to perform an action on the current pixel, knowing its *status*.

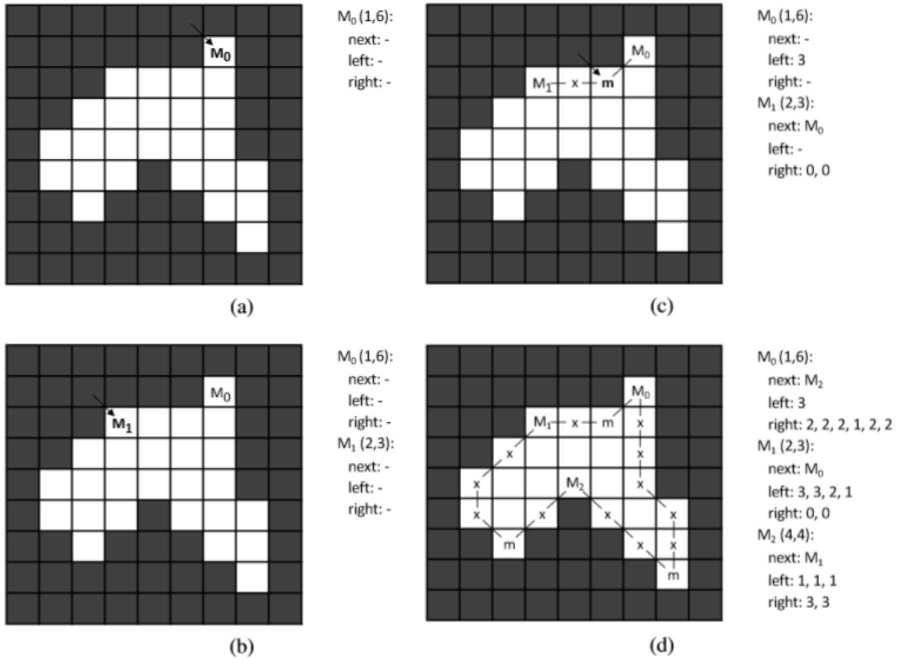
```
Input: Mv, Cv, pos, r, c, status
Procedure PerformAction():
    last_found_right ⟵ false
    pos ⟵ pos − CountHorizLinks(status)
    for type = 0 to 3 do
        if IsLeftLinkType(status, type) then
            Mv[Cv[pos]].left.push_back(link)
            pos ⟵ pos + 1
            last_found_right ⟵ false
        if IsRightLinkType(status, type) then
            Mv[Cv[pos]].right.push_back(link)
            pos ⟵ pos + 1
            last_found_right ⟵ true
    if IsInnerMinPoint(status) then
        Mv[Cv[pos − 2]].next ⟵ Cv[pos − 1]
        Cv.erase(from=pos − 2, to=pos)
    if IsOuterMinPoint(status) then
        Mv[Cv[pos − 1]].next ⟵ Cv[pos − 2]
        Cv.erase(from=pos − 2, to=pos)
    if IsOuterMaxPoint(status) then
        Mv.emplace_back(r, c)
        Cv.insert(at=pos, cnt=2, val=Mv.size)
        last_found_right ⟵ true
    if IsInnerMaxPoint(status) then
        Mv.emplace_back(r, c)
        if last_found_right then
            Cv.insert(at=pos − 1, cnt=2, val=Mv.size)
        else
            Cv.insert(at=pos, cnt=2, val=Mv.size)
```

Algorithm 2: Decision tree implementation. Border pixels check are omitted for readability purposes.

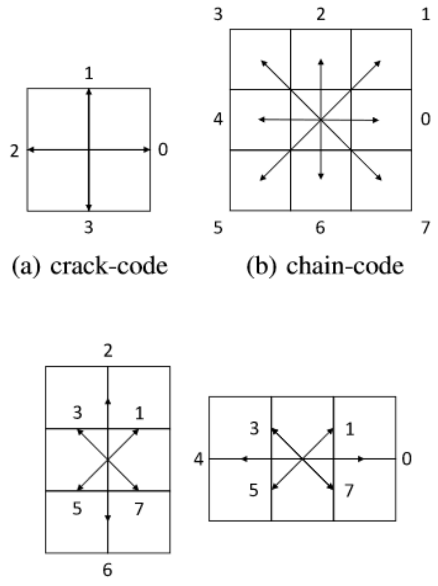
```
Input: I binary image; r, c row and column indexes.
Output: status of pixel I[r, c].
Function DecisionTree():
    if I[r, c] then                                  // x
        if I[r − 1, c] then                          // b
            if I[r, c − 1] then                      // d
                if I[r, c + 1] then                  // e
                    if I[r + 1, c] then              // g
                        return 1
                    else
                        if I[r + 1, c − 1] then      // f
                            return 2
                        else
                            return 51
        else
            // ... the rest of the tree
```

Algorithm 3: The complete Cederberg RC-code extraction algorithm.

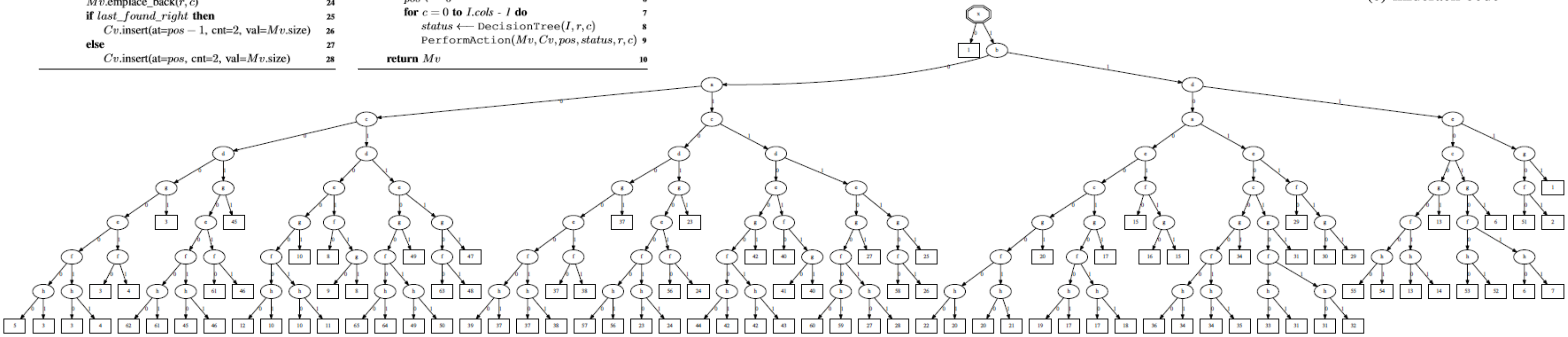
```
Input: I binary image.
Output: Mv vector of MaxPoints.
Function ExtractChainCode():
    Mv ⟵ vector < MaxPoint >
    Cv ⟵ vector < int >
    // Cv contains, for each chain, the index
       of its MaxPoint in Mv
    for r = 0 to I.rows - 1 do
        pos ⟵ 0
        for c = 0 to I.cols - 1 do
            status ⟵ DecisionTree(I, r, c)
            PerformAction(Mv, Cv, pos, status, r, c)
    return Mv
```



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



(a) crack-code (b) chain-code (c) midcrack-code



Algorithm 1: This algorithm provides the Cederberg RC-code action implementation, *i.e.* the instructions required to perform an action on the current pixel, knowing its *status*.

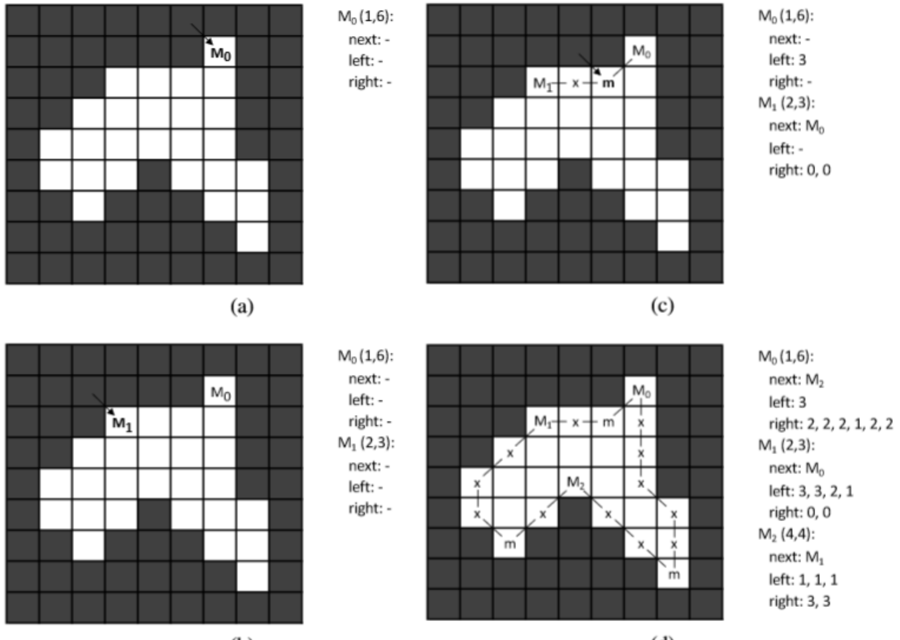
```
Input: Mv, Cv, pos, r, c, status
Procedure PerformAction():
    last_found_right ⟵ false
    pos ⟵ pos − CountHorizLinks(status)
    for type = 0 to 3 do
        if IsLeftLinkType(status, type) then
            Mv[Cv[pos]].left.push_back(link)
            pos ⟵ pos + 1
            last_found_right ⟵ false
        if IsRightLinkType(status, type) then
            Mv[Cv[pos]].right.push_back(link)
            pos ⟵ pos + 1
            last_found_right ⟵ true
    if IsInnerMinPoint(status) then
        Mv[Cv[pos − 2]].next ⟵ Cv[pos − 1]
        Cv.erase(from=pos − 2, to=pos)
    if IsOuterMinPoint(status) then
        Mv[Cv[pos − 1]].next ⟵ Cv[pos − 2]
        Cv.erase(from=pos − 2, to=pos)
    if IsOuterMaxPoint(status) then
        Mv.emplace_back(r, c)
        Cv.insert(at=pos, cnt=2, val=Mv.size)
        last_found_right ⟵ true
    if IsInnerMaxPoint(status) then
        Mv.emplace_back(r, c)
        if last_found_right then
            Cv.insert(at=pos − 1, cnt=2, val=Mv.size)
        else
            Cv.insert(at=pos, cnt=2, val=Mv.size)
```

Algorithm 2: Decision tree implementation. Border pixels check are omitted for readability purposes.

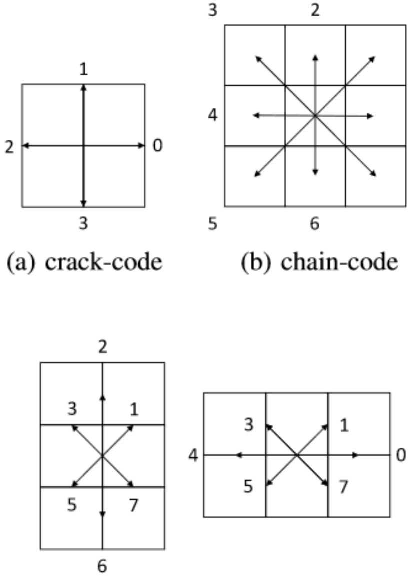
```
Input: I binary image; r, c row and column indexes.
Output: status of pixel I[r, c].
Function DecisionTree():
    if I[r, c] then                          // x
        if I[r − 1, c] then                  // b
            if I[r, c − 1] then              // d
                if I[r, c + 1] then          // e
                    if I[r + 1, c] then      // g
                        return 1
                    else
                        if I[r + 1, c − 1] then   // f
                            return 2
                        else
                            return 51
        else
            // ... the rest of the tree
```

Algorithm 3: The complete Cederberg RC-code extraction algorithm.

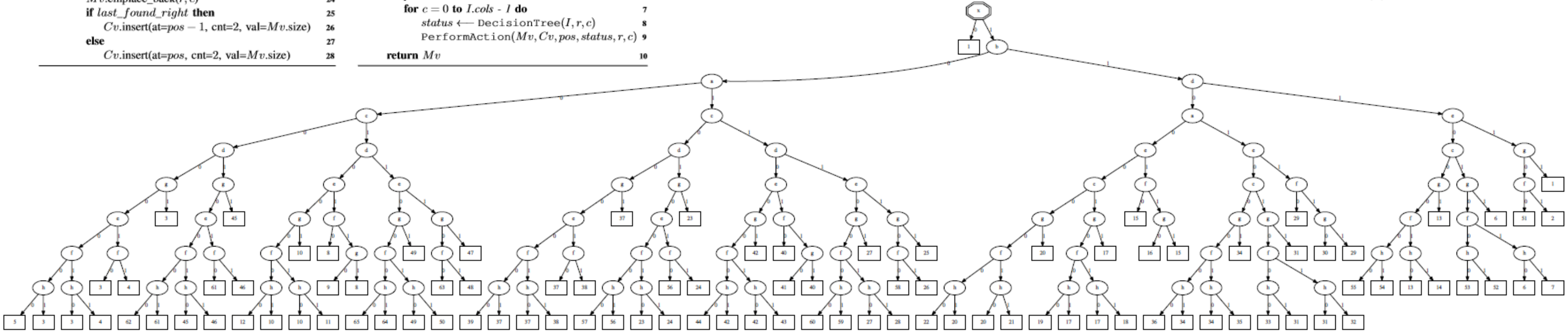
```
Input: I binary image.
Output: Mv vector of MaxPoints.
Function ExtractChainCode():
    Mv ⟵ vector < MaxPoint >
    Cv ⟵ vector < int >
    // Cv contains, for each chain, the index
       of its MaxPoint in Mv
    for r = 0 to I.rows - 1 do
        pos ⟵ 0
        for c = 0 to I.cols - 1 do
            status ⟵ DecisionTree(I, r, c)
            PerformAction(Mv, Cv, pos, status, r, c)
    return Mv
```

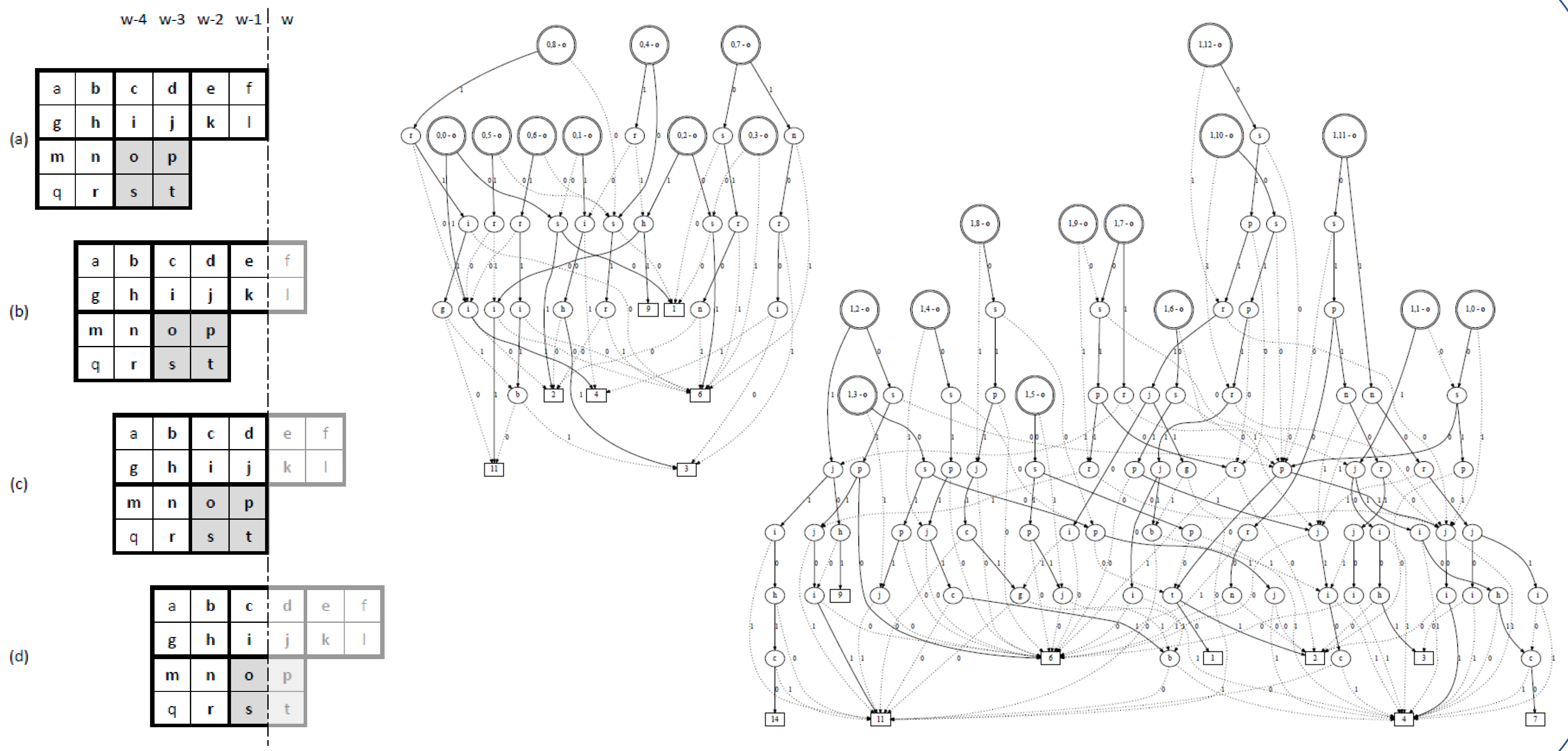


(a) (b) (c) (d)

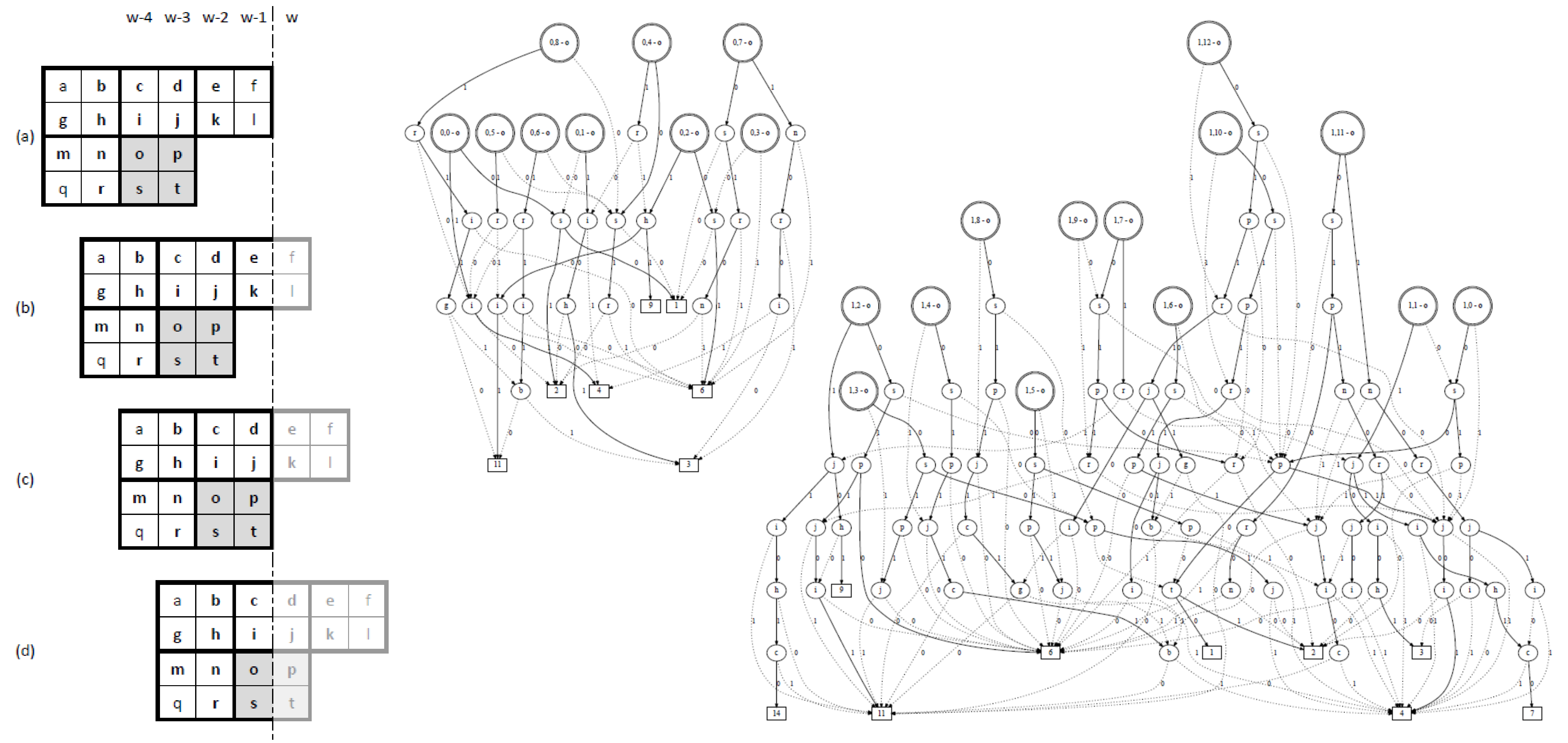


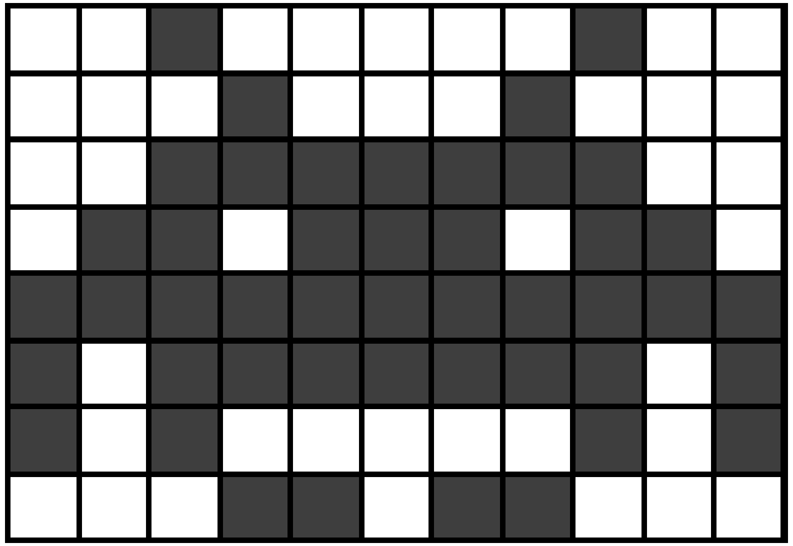
(a) crack-code (b) chain-code (c) midcrack-code



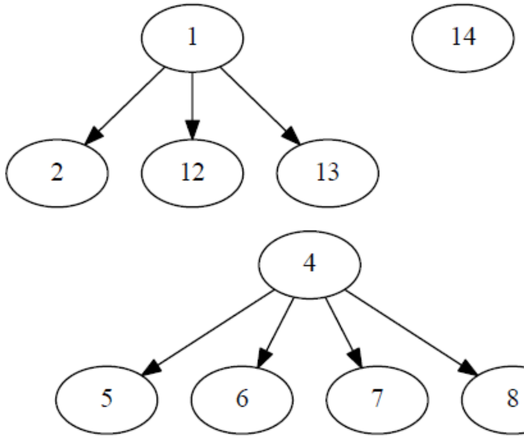


w-4 w-3 w-2 w-1 w
(a)
(b)
(c)
(d)


w-4 w-3 w-2 w-1 w
(a)
(b)
(c)
(d)

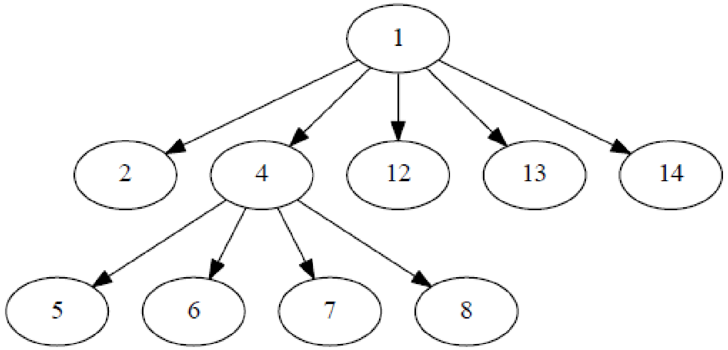


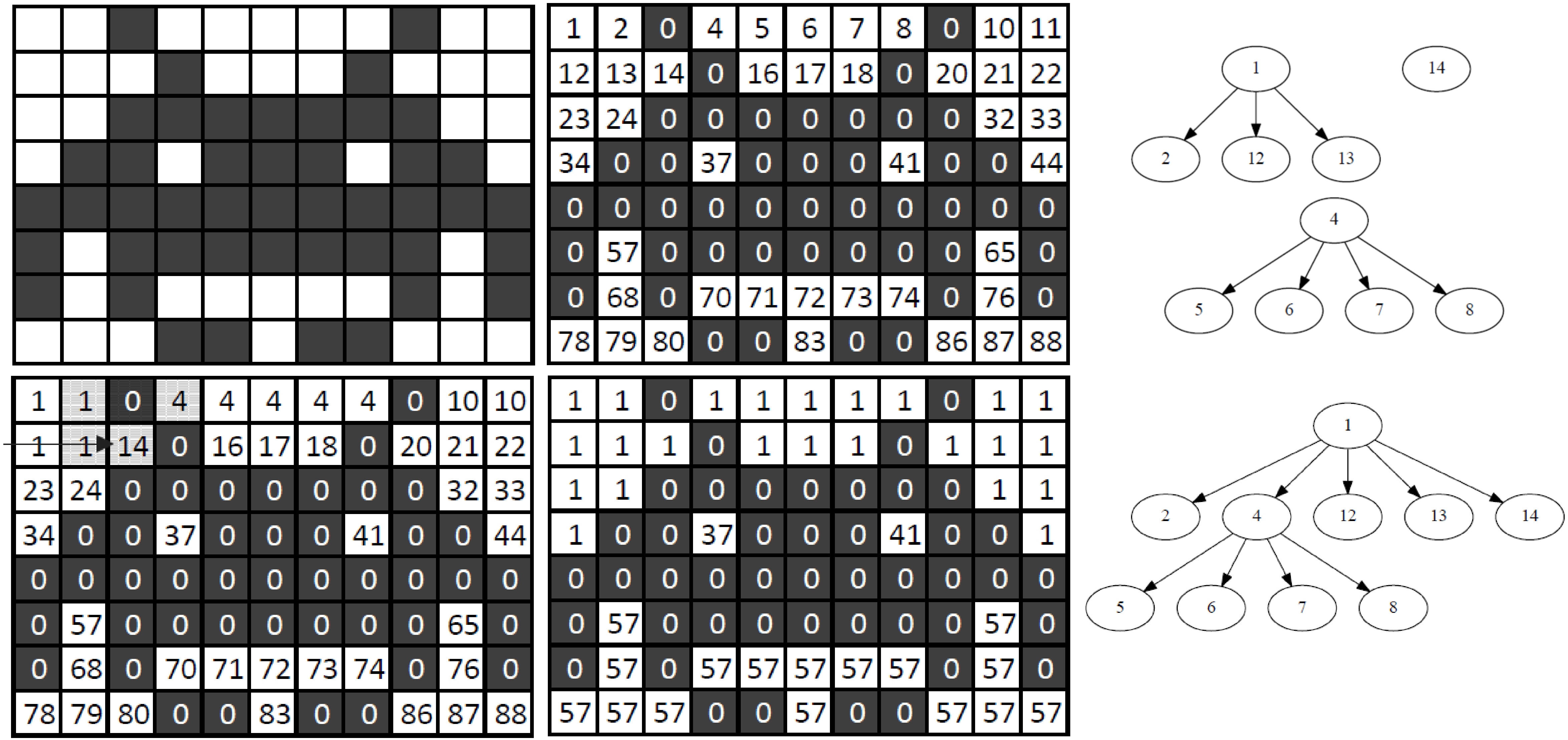
1	2	0	4	5	6	7	8	0	10	11
12	13	14	0	16	17	18	0	20	21	22
23	24	0	0	0	0	0	0	0	32	33
34	0	0	37	0	0	0	41	0	0	44
0	0	0	0	0	0	0	0	0	0	0
0	57	0	0	0	0	0	0	0	65	0
0	68	0	70	71	72	73	74	0	76	0
78	79	80	0	0	83	0	0	86	87	88

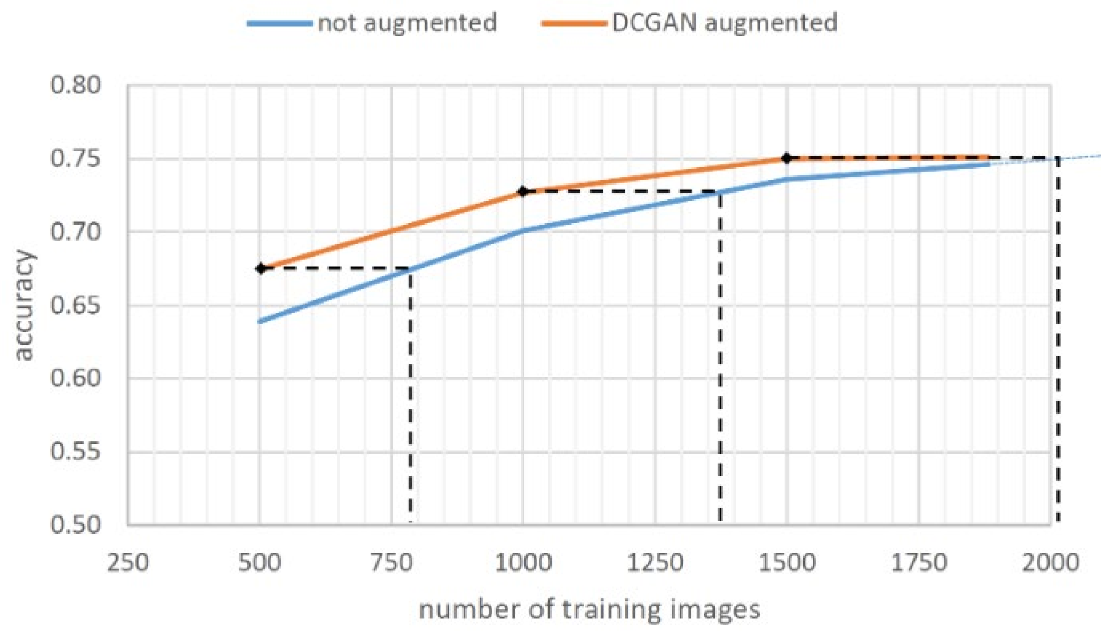


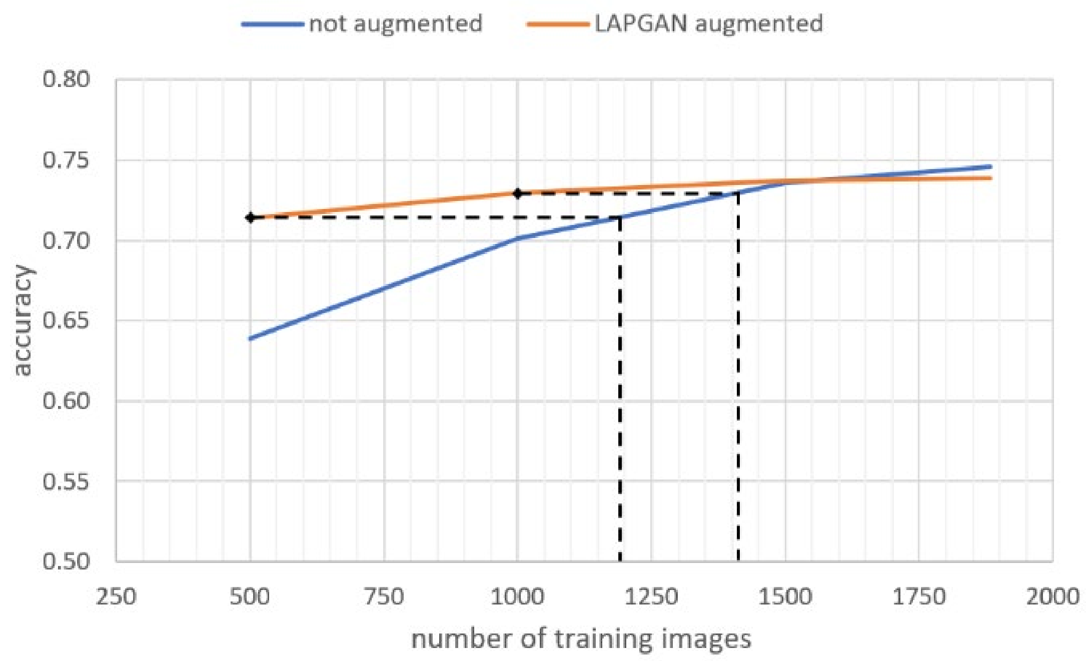
1	1	0	4	4	4	4	4	0	10	10
1	1	14	0	16	17	18	0	20	21	22
23	24	0	0	0	0	0	0	0	32	33
34	0	0	37	0	0	0	41	0	0	44
0	0	0	0	0	0	0	0	0	0	0
0	57	0	0	0	0	0	0	0	65	0
0	68	0	70	71	72	73	74	0	76	0
78	79	80	0	0	83	0	0	86	87	88

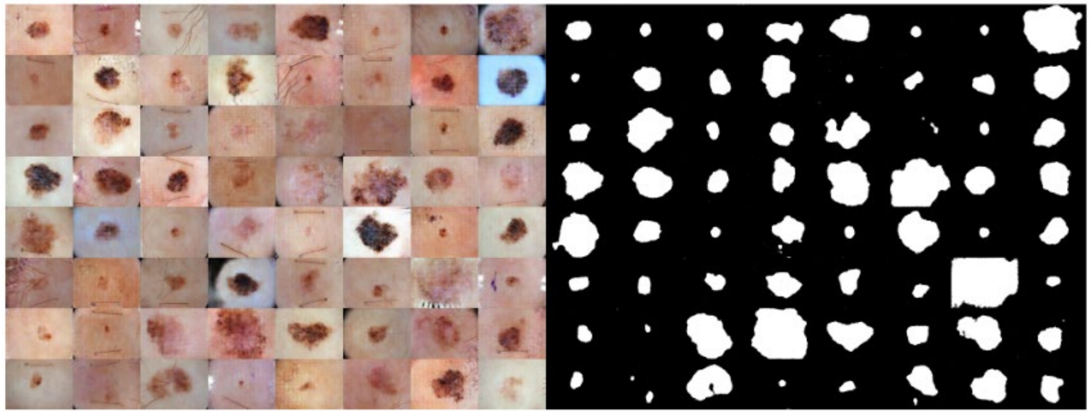
1	1	0	1	1	1	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1	1
1	1	0	0	0	0	0	0	0	1	1
1	0	0	37	0	0	0	41	0	0	1
0	0	0	0	0	0	0	0	0	0	0
0	57	0	0	0	0	0	0	0	57	0
0	57	0	57	57	57	57	57	0	57	0
57	57	57	0	0	57	0	0	57	57	57

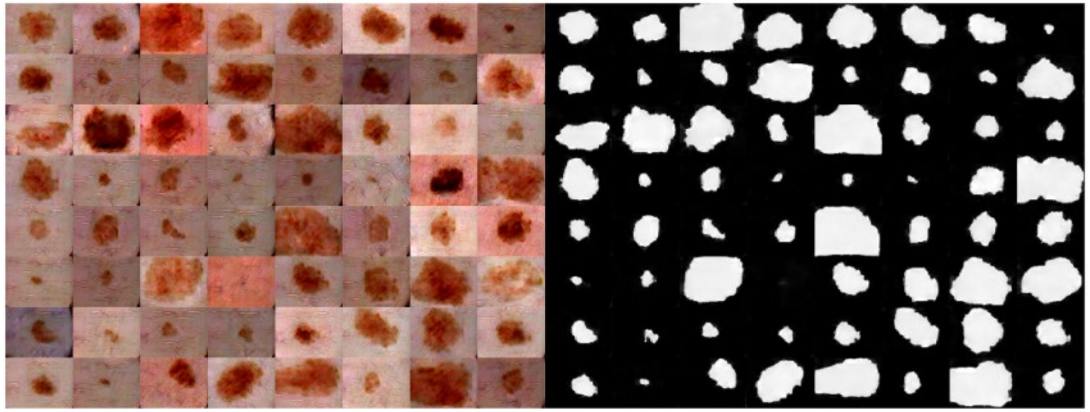


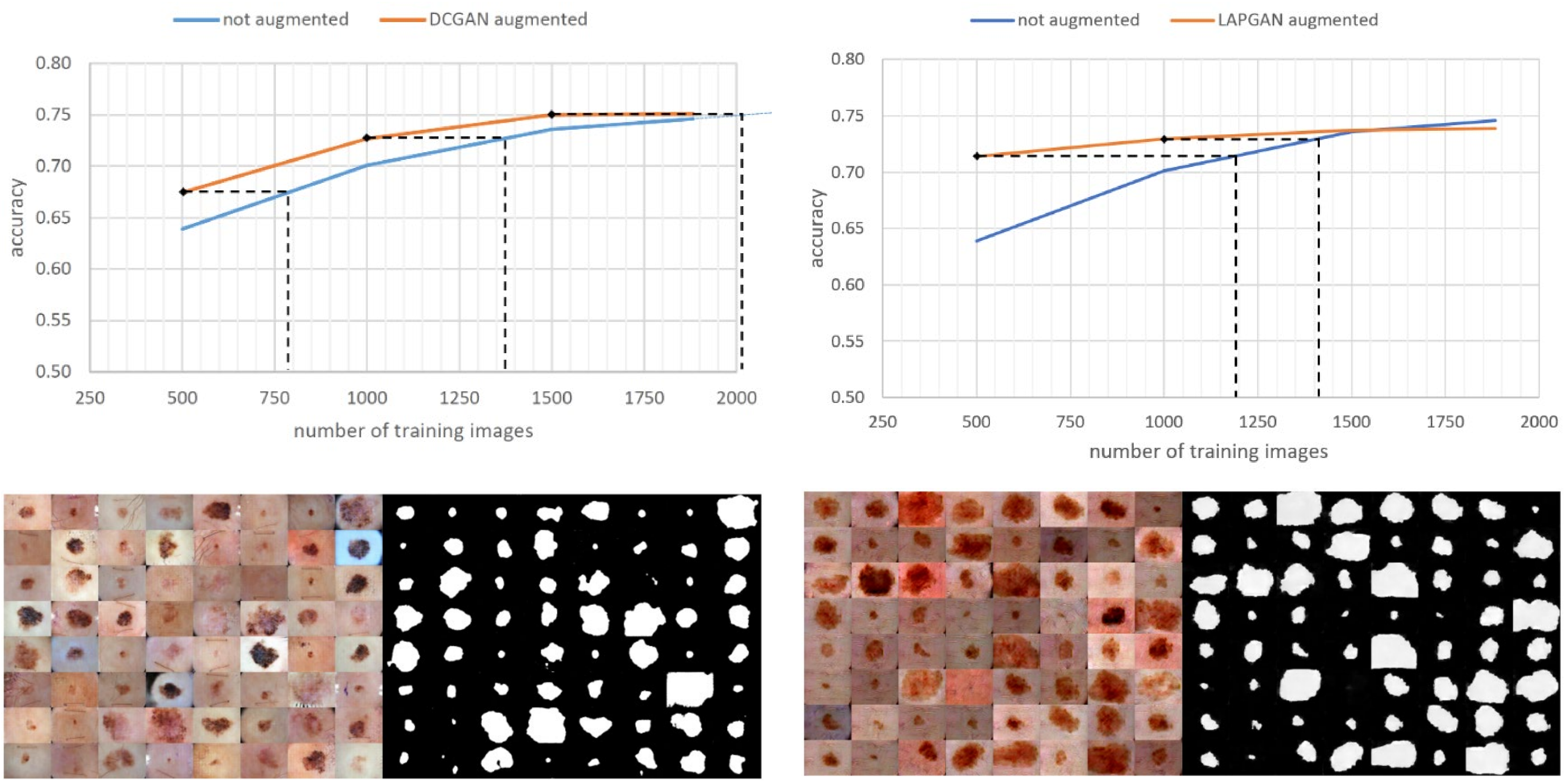

1 2 0 4 5 6 7 8 0 10 11
12 13 14 0 16 17 18 0 20 21 22
23 24 0 0 0 0 0 0 0 32 33
34 0 0 37 0 0 0 41 0 0 44
0 0 0 0 0 0 0 0 0 0 0
0 57 0 0 0 0 0 0 0 65 0
0 68 0 70 71 72 73 74 0 76 0
78 79 80 0 0 83 0 0 86 87 88
1
2
12
13
14
4
5
6
7
8
1 1 0 4 4 4 4 4 0 10 10
1 1 14 0 16 17 18 0 20 21 22
23 24 0 0 0 0 0 0 0 32 33
34 0 0 37 0 0 0 41 0 0 44
0 0 0 0 0 0 0 0 0 0 0
0 57 0 0 0 0 0 0 0 65 0
0 68 0 70 71 72 73 74 0 76 0
78 79 80 0 0 83 0 0 86 87 88
1 1 0 1 1 1 1 1 0 1 1
1 1 1 0 1 1 1 0 1 1 1
1 1 0 0 0 0 0 0 0 1 1
1 0 0 37 0 0 0 41 0 0 1
0 0 0 0 0 0 0 0 0 0 0
0 57 0 0 0 0 0 0 0 57 0
0 57 0 57 57 57 57 57 0 57 0
57 57 57 0 0 57 0 0 57 57 57
1
2
4
12
13
14
5
6
7
8

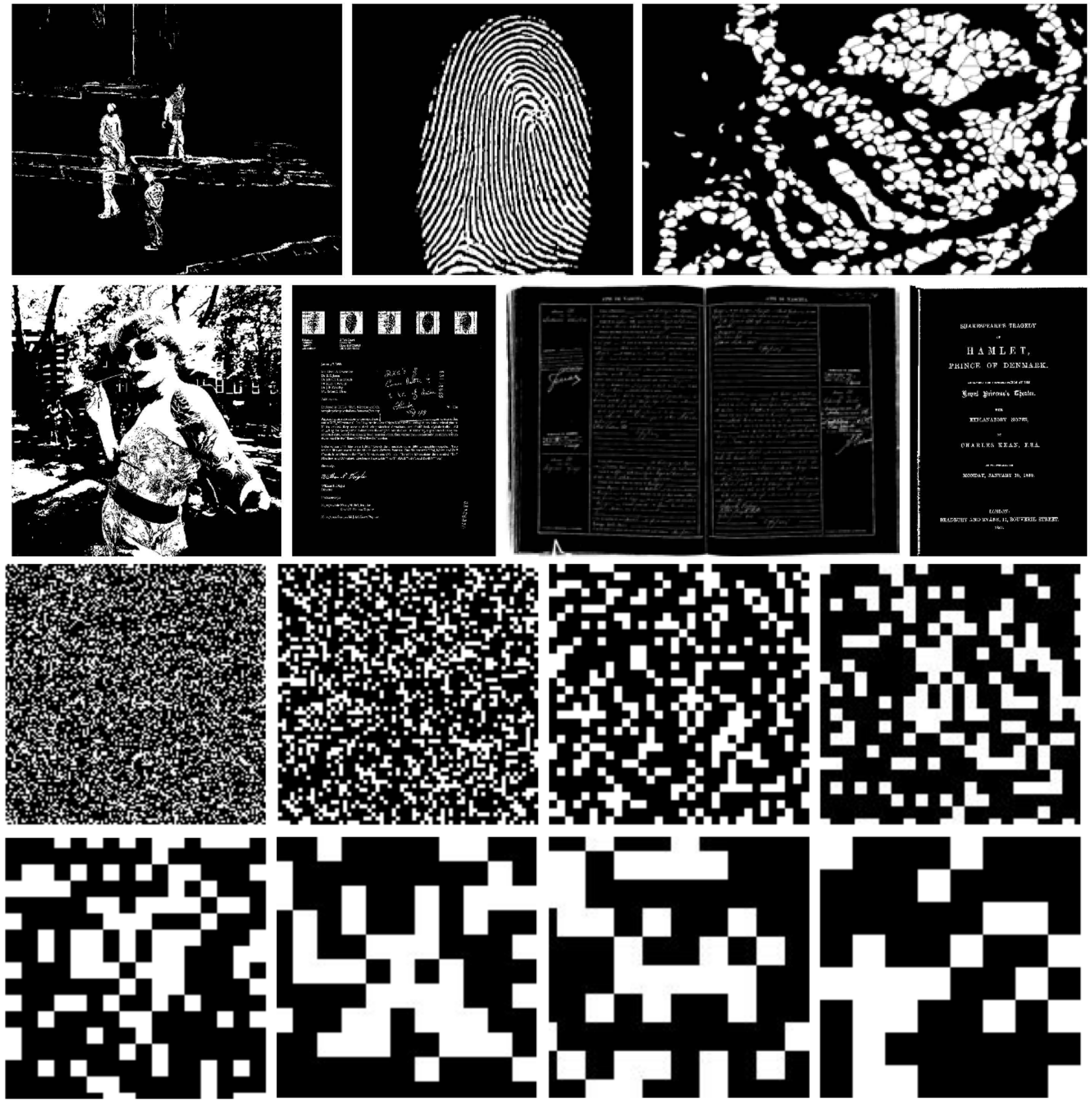

not augmented
DCGAN augmented
0.80
0.75
0.70
0.65
0.60
0.55
0.50
accuracy
250 500 750 1000 1250 1500 1750 2000
number of training images

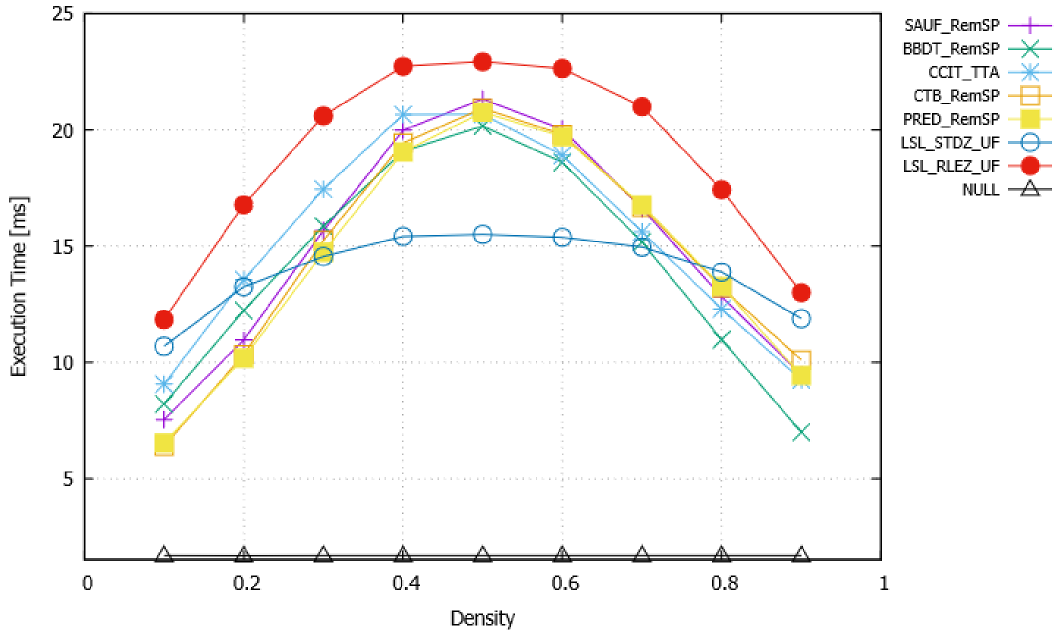

not augmented
LAPGAN augmented
0.80
0.75
0.70
0.65
0.60
0.55
0.50
accuracy
250 500 750 1000 1250 1500 1750 2000
number of training images



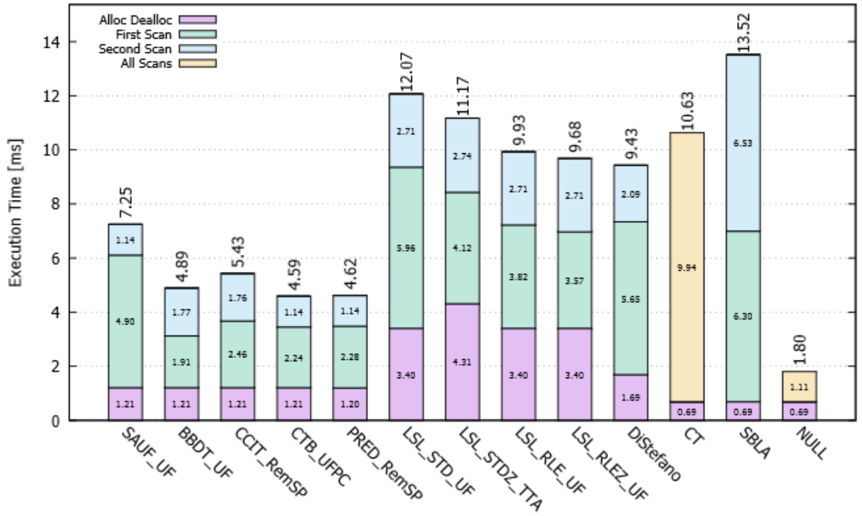



not augmented
DCGAN augmented
0.80
0.75
0.70
0.65
0.60
0.55
0.50
accuracy
250
500
750
1000
1250
1500
1750
2000
number of training images
not augmented
LAPGAN augmented
0.80
0.75
0.70
0.65
0.60
0.55
0.50
accuracy
250
500
750
1000
1250
1500
1750
2000
number of training images

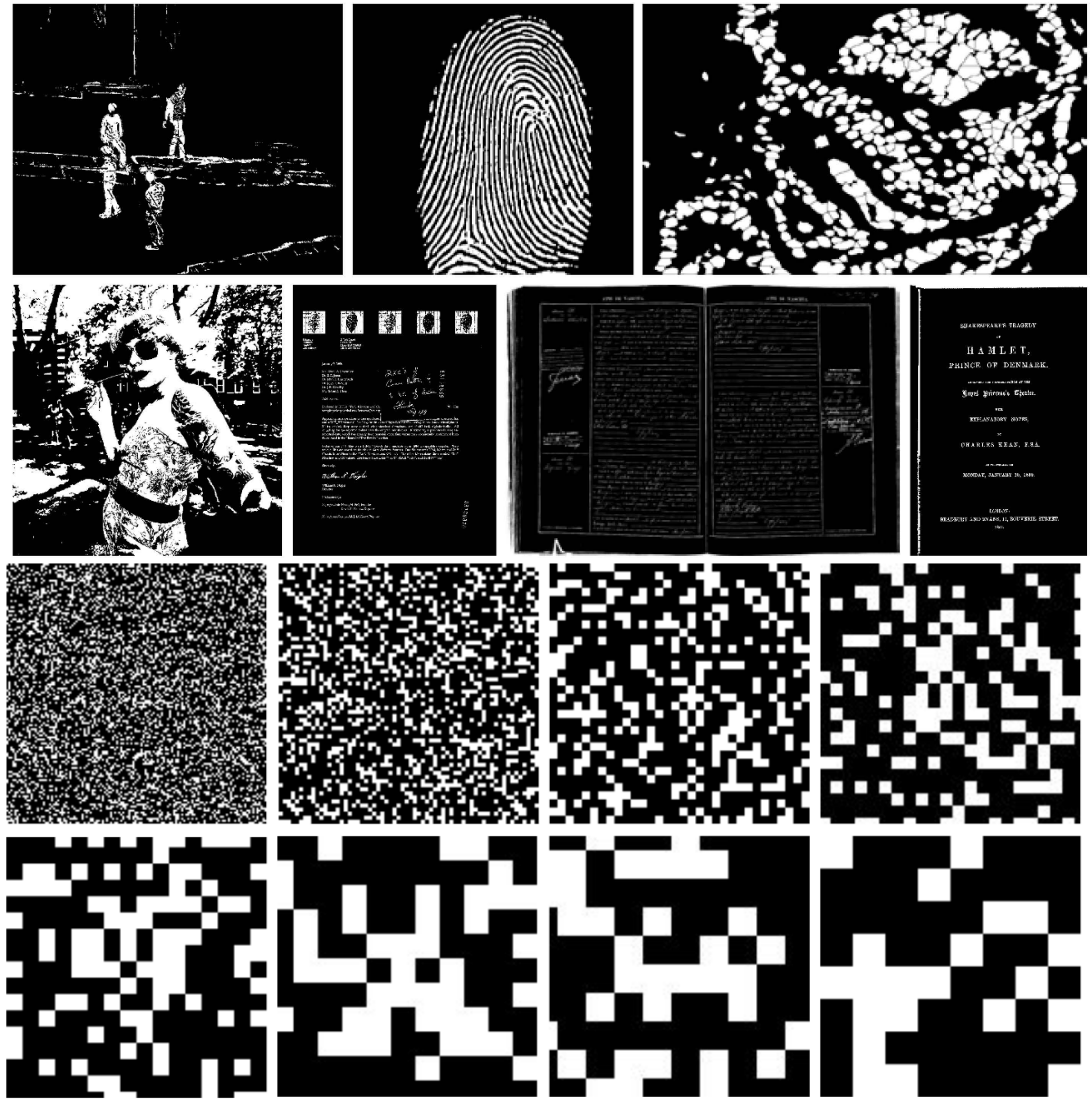


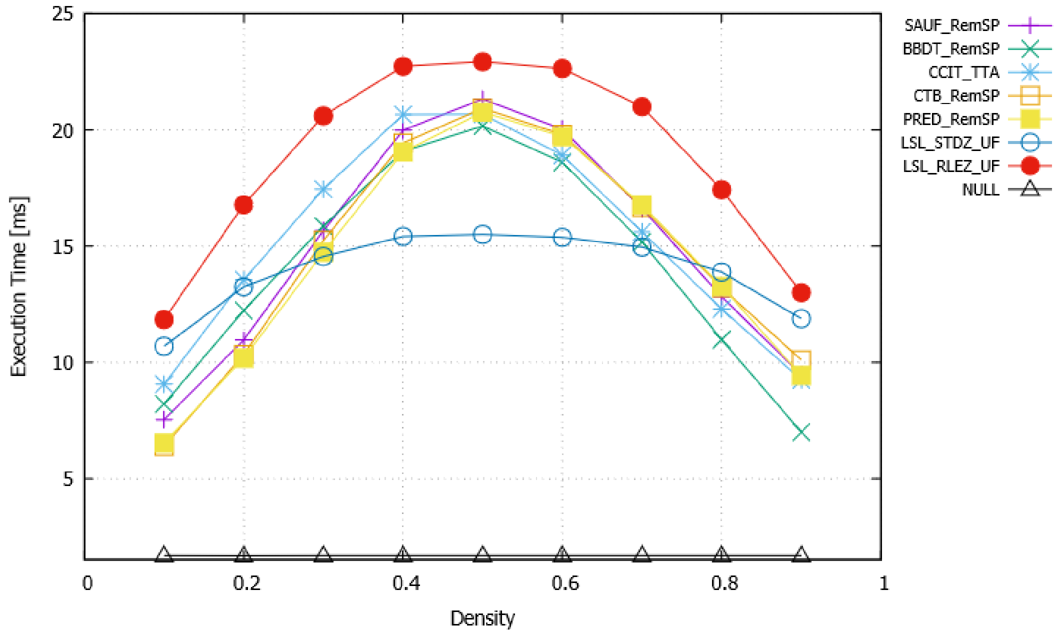


SAUF_RemSP
BBDT_RemSP
CCIT_TTA
CTB_RemSP
PRED_RemSP
LSL_STDZ_UF
LSL_RLEZ_UF
NULL
Execution Time [ms]
25
20
15
10
5
0
0.2
0.4
0.6
0.8
1
Density

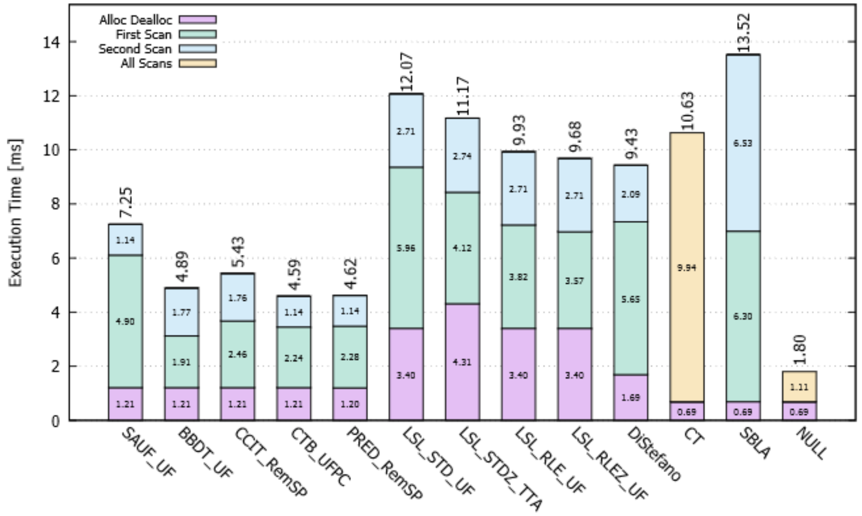


Alloc Dealloc
First Scan
Second Scan
All Scans
Execution Time [ms]
14
12
10
8
6
4
2
0
7.25
1.14
4.90
1.21
4.89
1.77
1.91
1.21
5.43
1.76
2.46
1.21
4.59
1.14
2.24
1.21
4.62
1.14
2.28
1.20
12.07
2.71
5.96
3.40
11.17
2.74
4.12
4.31
9.93
2.71
3.82
3.40
9.68
2.71
3.57
3.40
9.43
2.09
5.65
1.69
10.63
9.94
0.69
13.52
6.53
6.30
0.69
1.80
1.11
0.69
SAUF_UF
BBDT_UF
CCIT_RemSP
CTB_UFPC
PRED_RemSP
LSL_STD_UF
LSL_STDZ_TTA
LSL_RLE_UF
LSL_RLEZ_UF
DiStefano
CT
SBLA
NULL

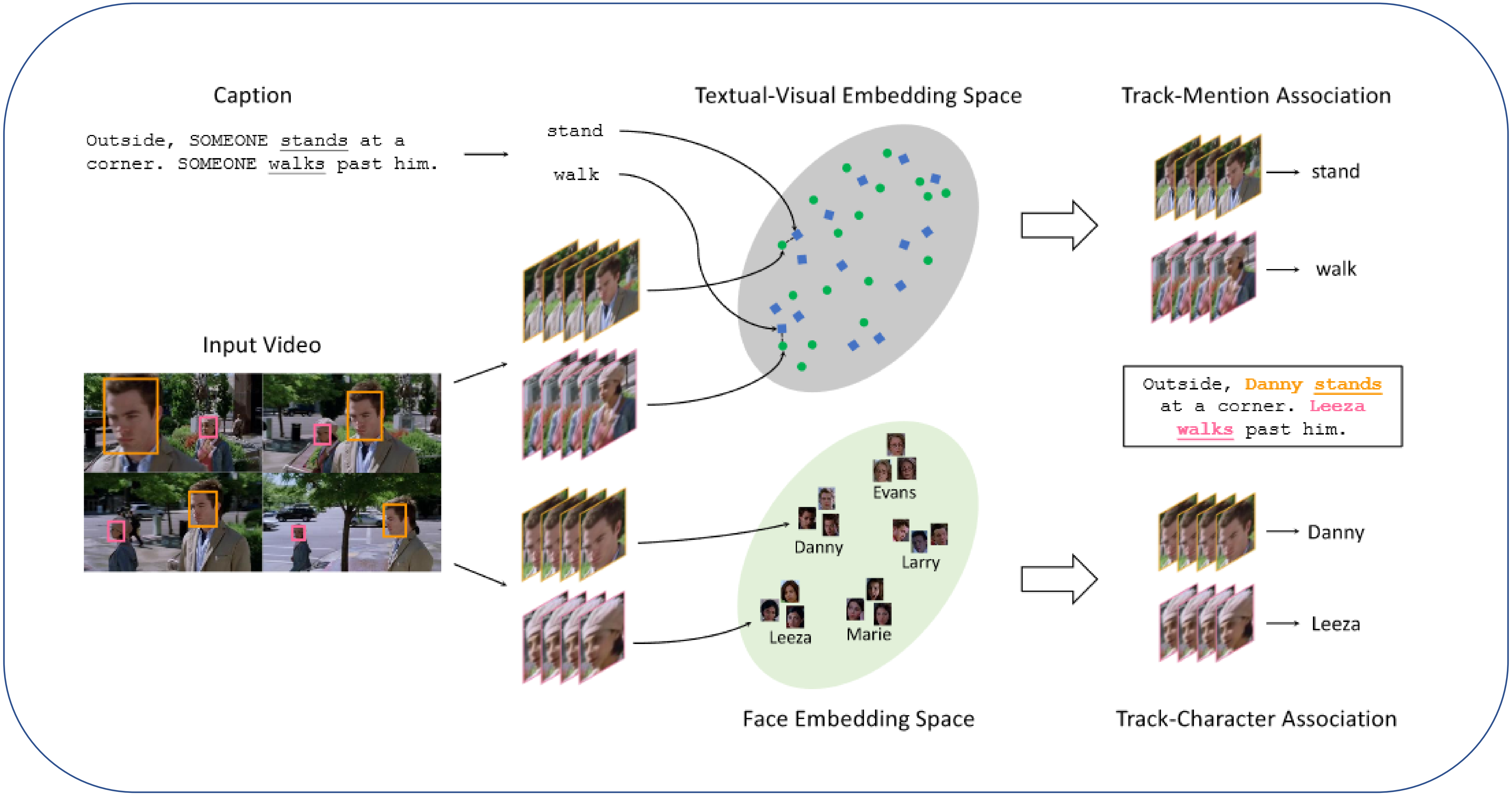




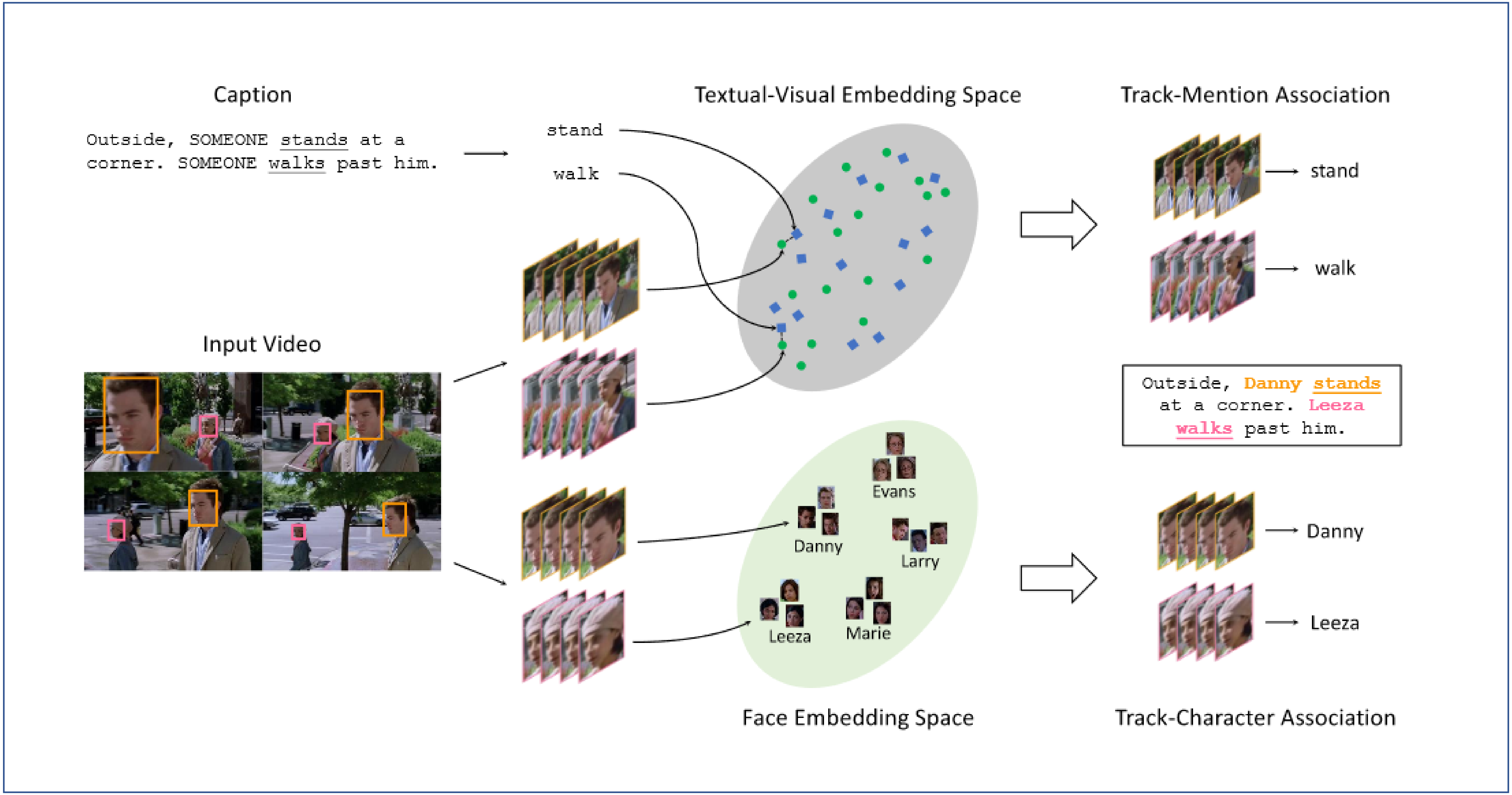
Execution Time [ms]
Density
SAUF_RemSP
BBDT_RemSP
CCIT_TTA
CTB_RemSP
PRED_RemSP
LSL_STDZ_UF
LSL_RLEZ_UF
NULL



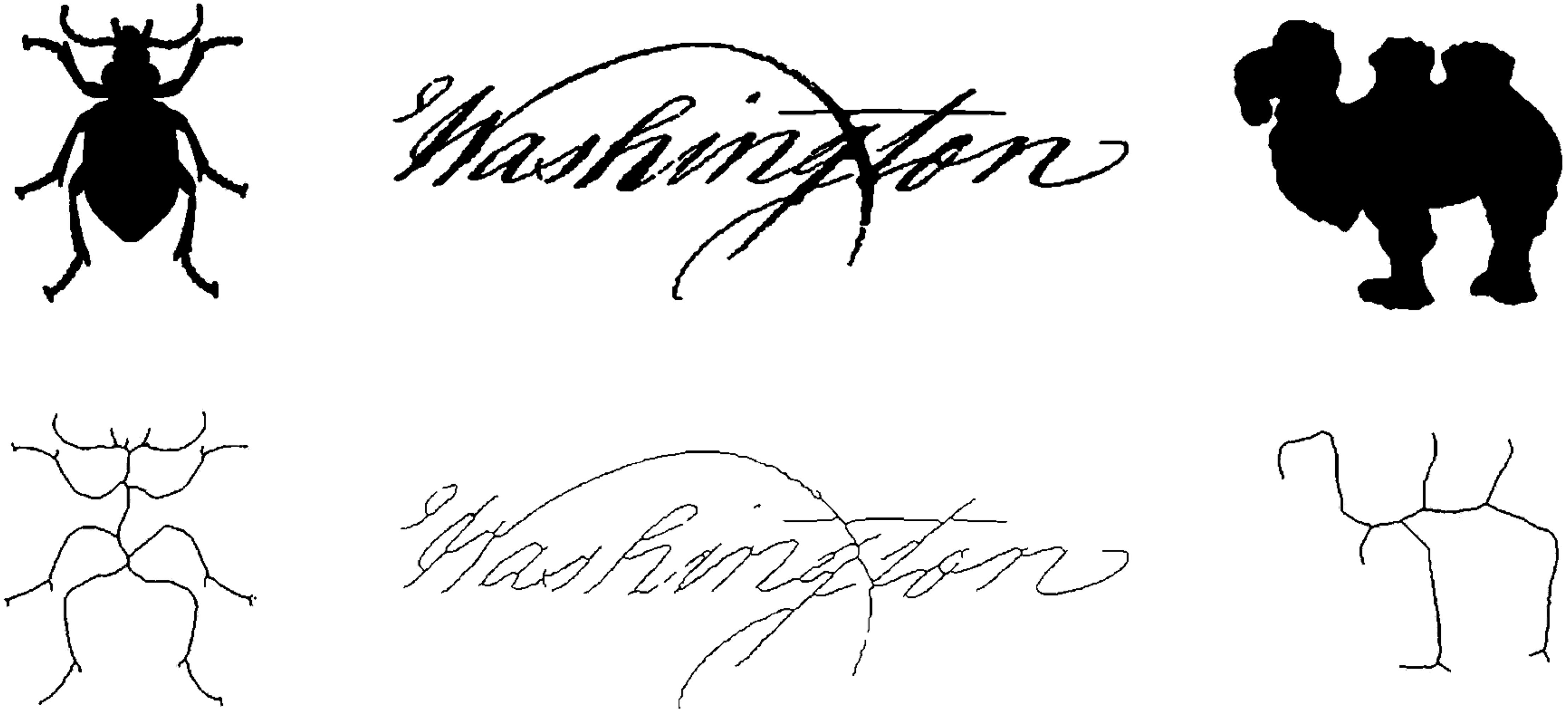
Alloc Dealloc
First Scan
Second Scan
All Scans
Execution Time [ms]
7.25
4.89
5.43
4.59
4.62
12.07
11.17
9.93
9.68
9.43
10.63
13.52
1.80
SAUF_UF
BBDT_UF
CCIT_RemSP
CTB_UFPC
PRED_RemSP
LSL_STD_UF
LSL_STDZ_TTA
LSL_RLE_UF
LSL_RLEZ_UF
DiStefano
CT
SBLA
NULL

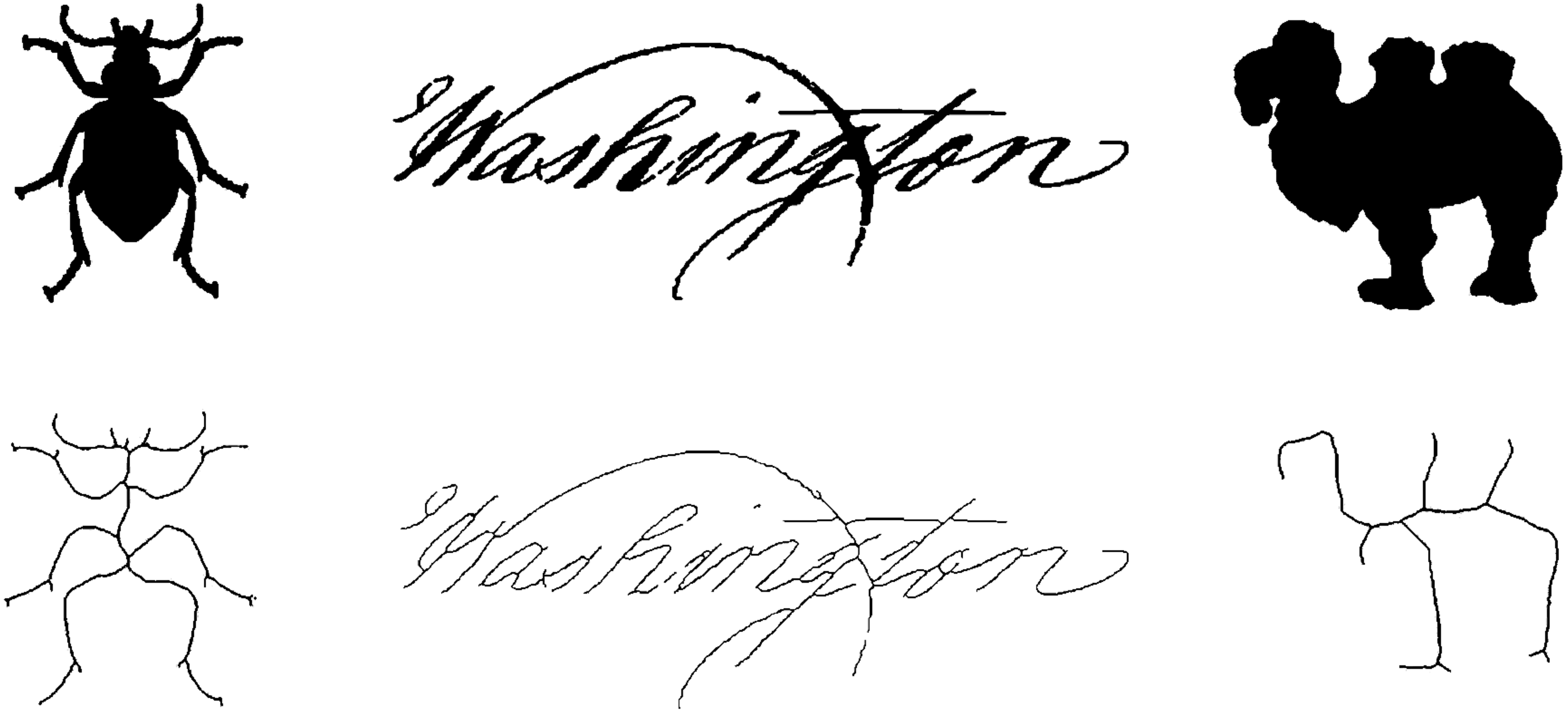


Caption
Outside, SOMEONE stands at a corner. SOMEONE walks past him.
stand
walk
Textual-Visual Embedding Space
Track-Mention Association
stand
walk
Input Video
Outside, Danny stands at a corner. Leeza walks past him.
Evans
Danny
Larry
Leeza
Marie
Danny
Leeza
Face Embedding Space
Track-Character Association

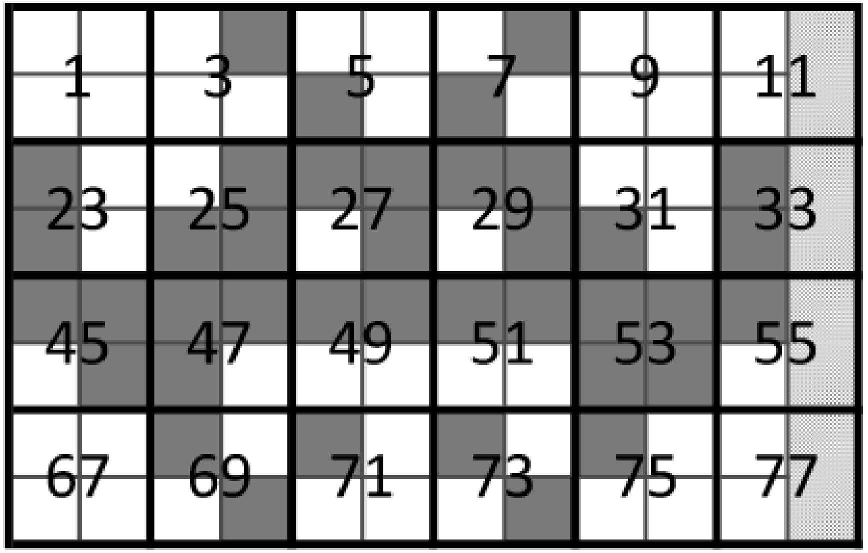


Caption
Outside, SOMEONE stands at a corner. SOMEONE walks past him.
Textual-Visual Embedding Space
stand
walk
Track-Mention Association
stand
walk
Outside, Danny stands at a corner. Leeza walks past him.
Input Video
Evans
Danny
Larry
Leeza
Marie
Danny
Leeza
Face Embedding Space
Track-Character Association

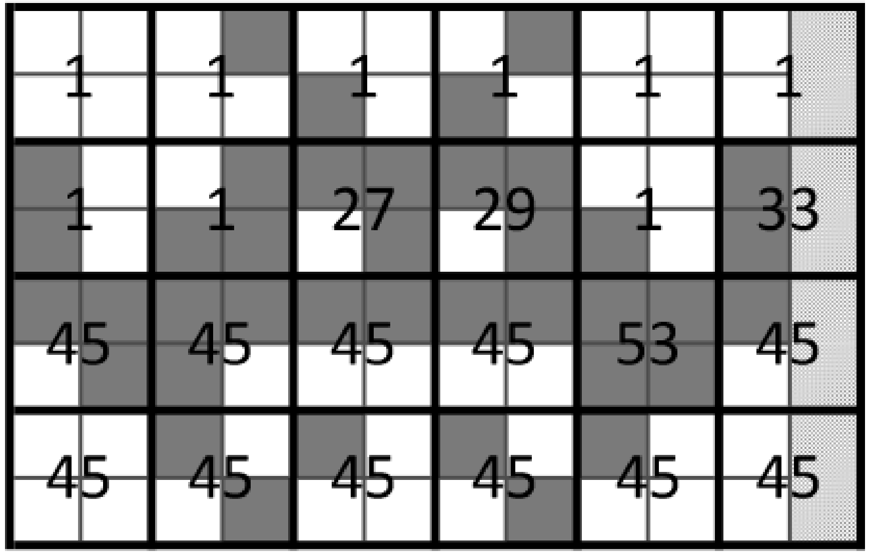

Washington


Washington
Washington

	3DPeS	*Fingerprints*	*Medical*	*MIRflickr*	*Tobacco800*	*XDOCS*
BUF*	**0.512**	**0.441**	**1.313**	**0.495**	**3.268**	**12.088**
BE	1.517	1.164	2.730	1.165	5.966	20.278
UF	0.594	0.529	2.040	0.659	4.304	17.316
OLE	1.211	1.128	3.013	1.281	8.173	35.242
KE	0.568	0.481	1.622	0.526	3.978	15.432

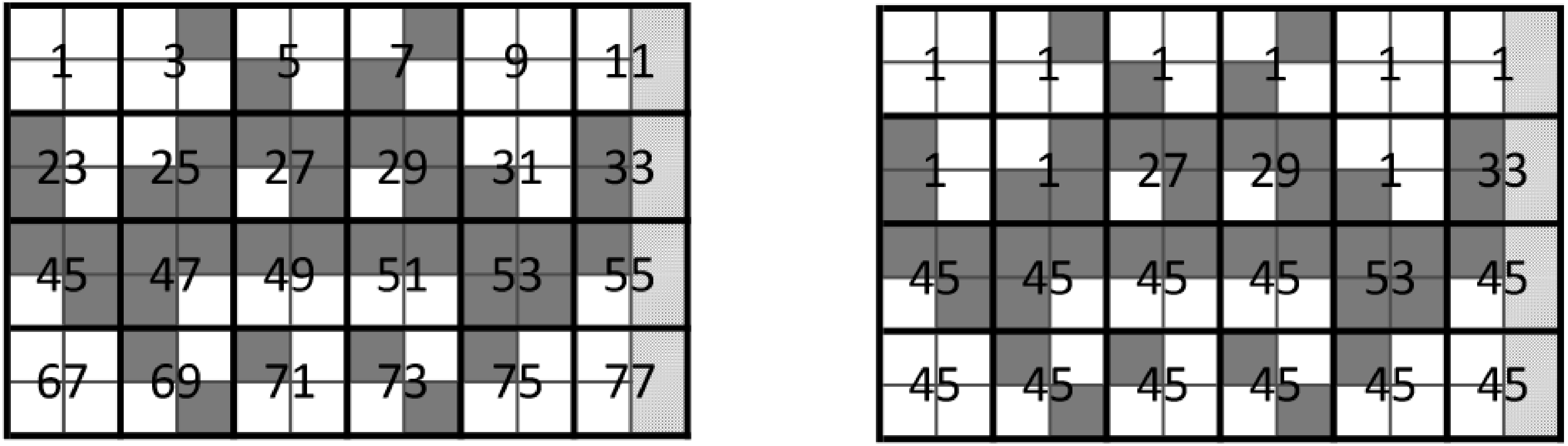


(a) Temporary Labels



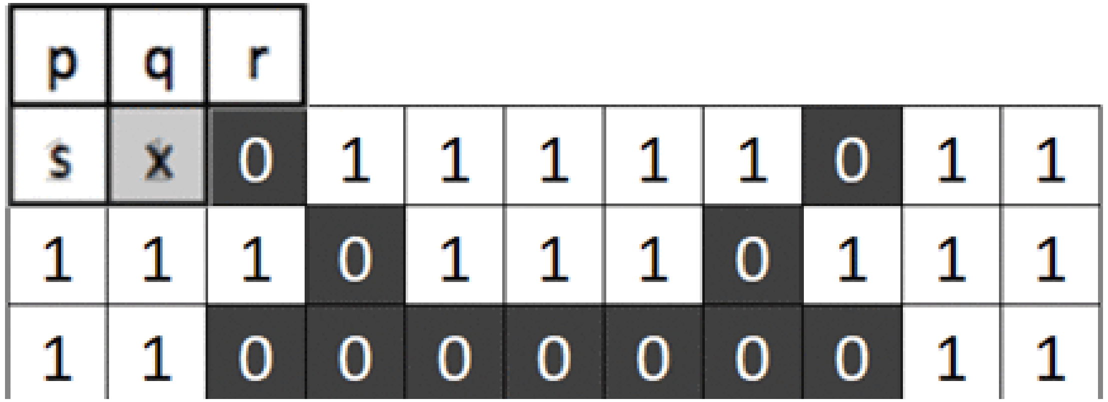
(b) Final Labels

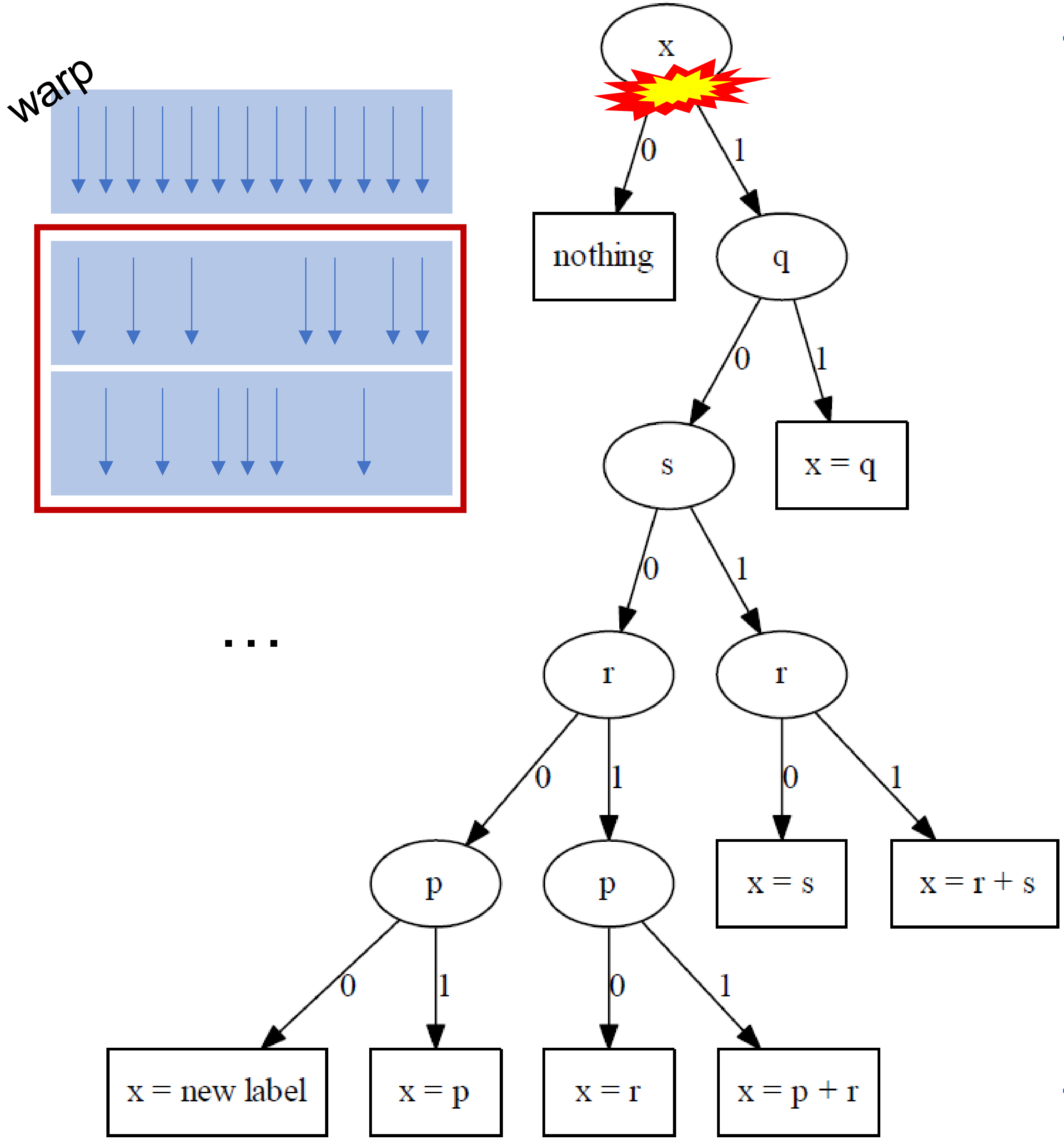
	3DPeS	*Fingerprints*	*Medical*	*MIRflickr*	*Tobacco800*	*XDOCS*
BUF*	**0.512**	**0.441**	**1.313**	**0.495**	**3.268**	**12.088**
BE	1.517	1.164	2.730	1.165	5.966	20.278
UF	0.594	0.529	2.040	0.659	4.304	17.316
OLE	1.211	1.128	3.013	1.281	8.173	35.242
KE	0.568	0.481	1.622	0.526	3.978	15.432

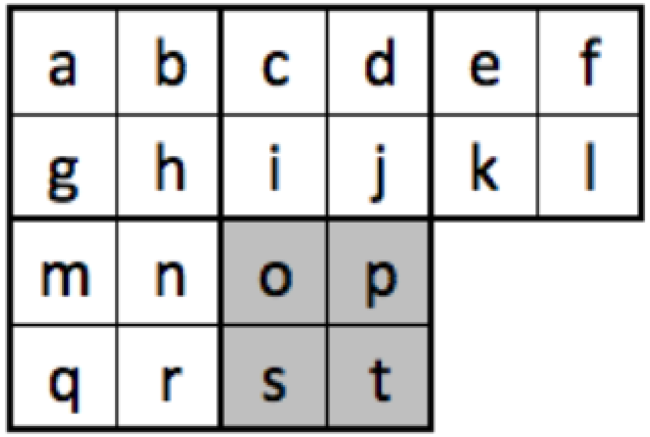


(a) Temporary Labels

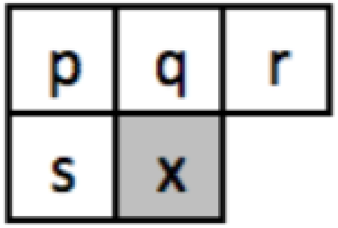
(b) Final Labels



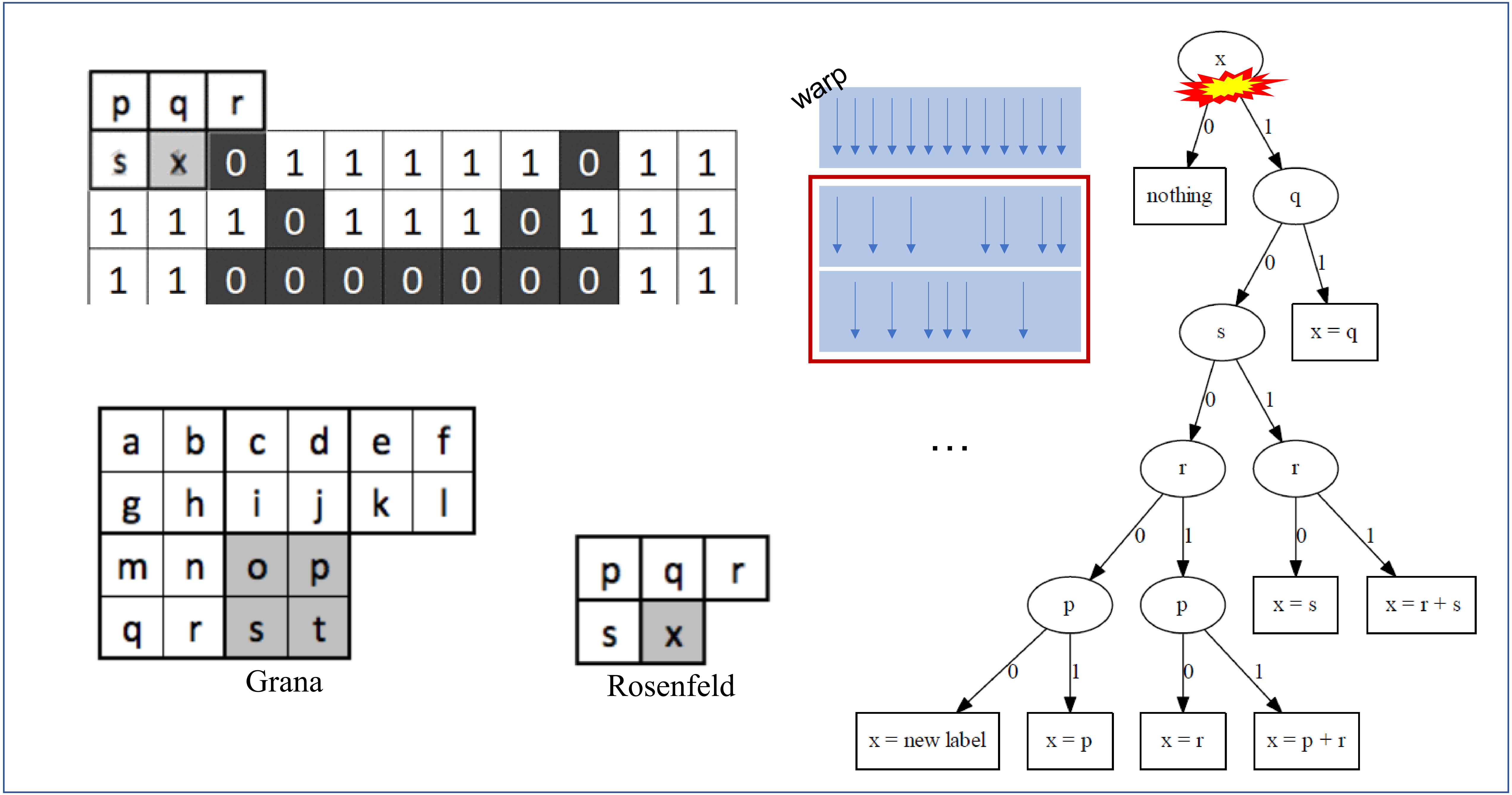




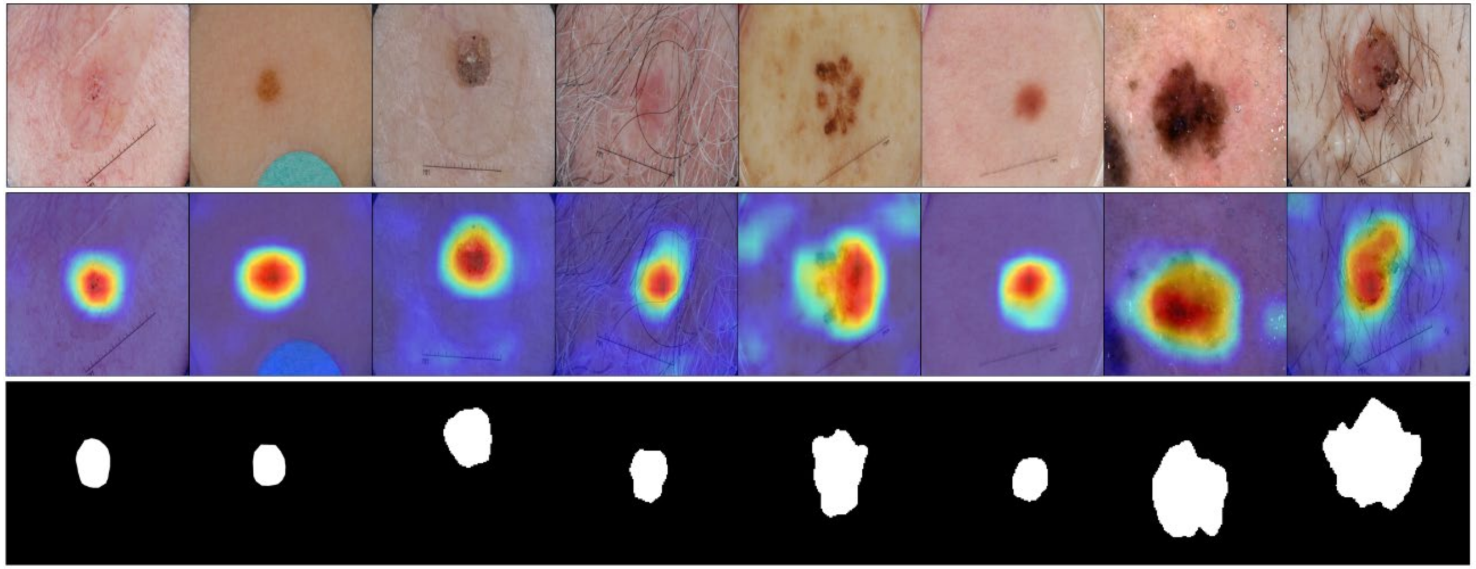
Grana



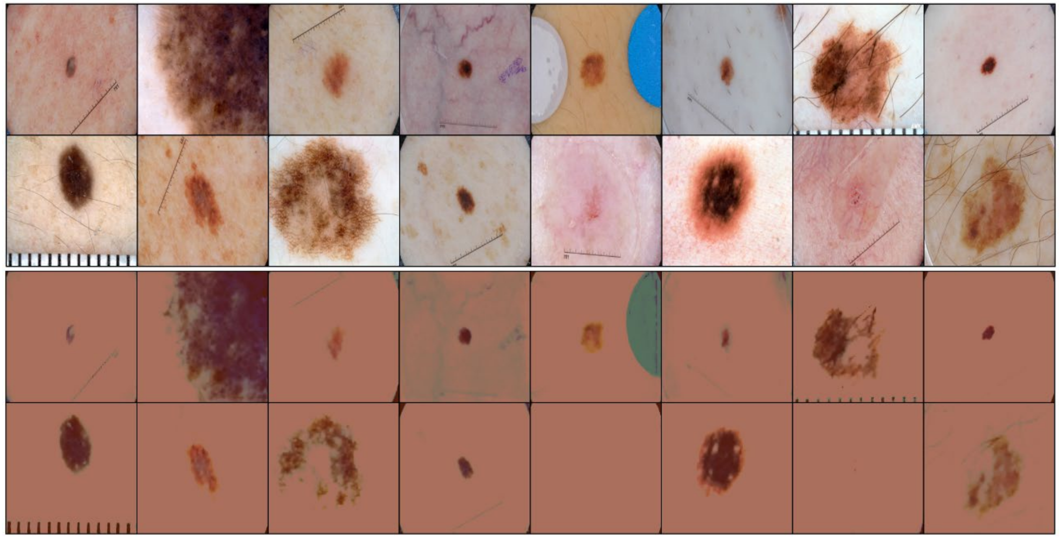
Rosenfeld

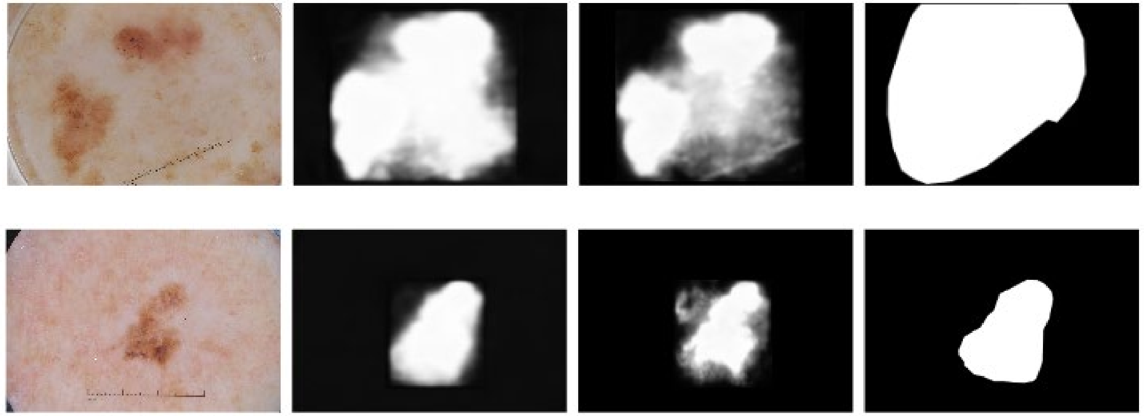


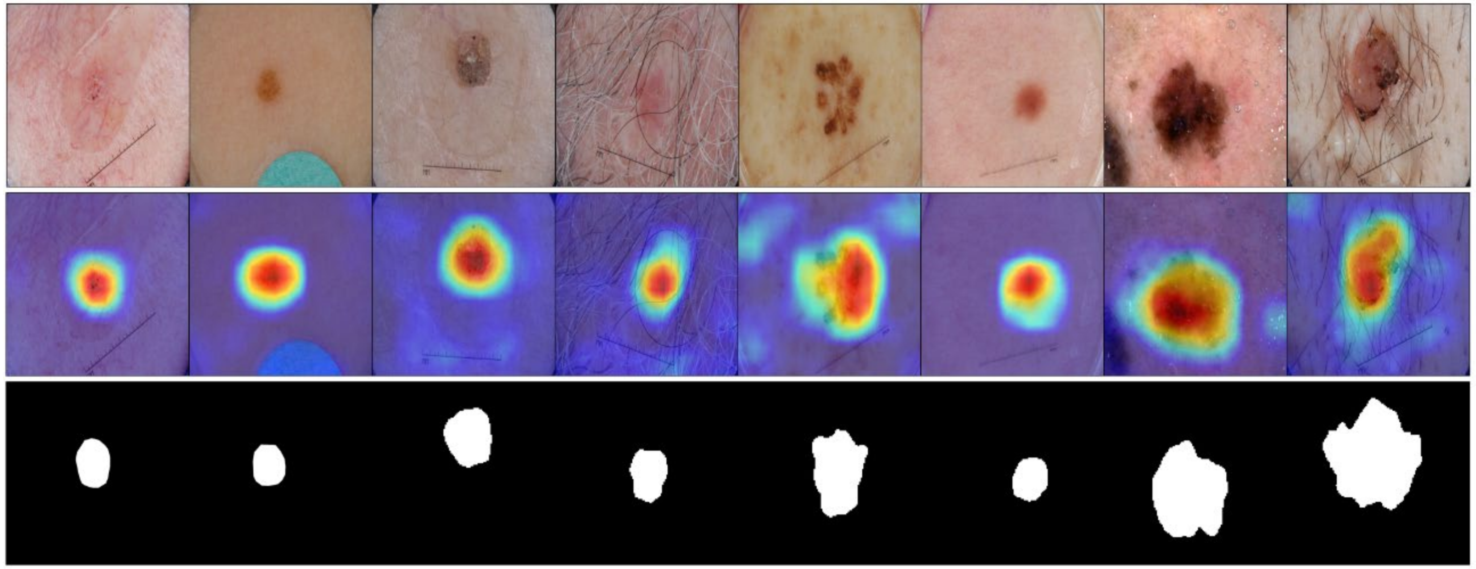
p q r
s x 0 1 1 1 1 1 0 1 1
1 1 1 0 1 1 1 0 1 1 1
1 1 0 0 0 0 0 0 0 1 1
warp
...
a b c d e f
g h i j k l
m n o p
q r s t
Grana
p q r
s x
Rosenfeld
x
nothing
q
x = q
s
r
r
x = s
x = r + s
p
p
x = new label
x = p
x = r
x = p + r



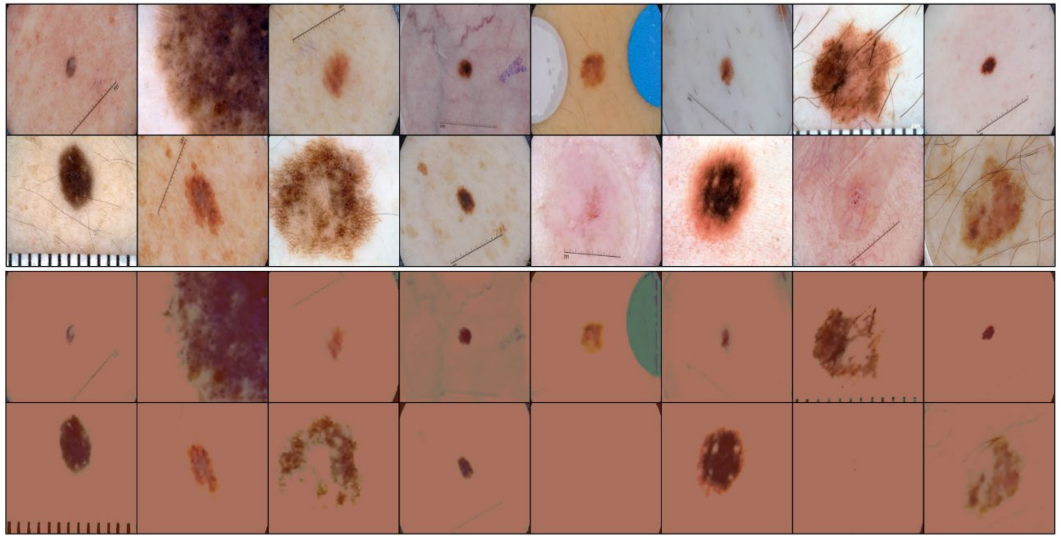
Method	Test IoU	Test TIoU
Ours (ensemble)	**0.850**	0.827
SegAN [35]	0.785	—
GAN Augmented [27]	0.781	—
DCL-PSI [2]	0.777	—
DeepLabv3+* [7]	0.769	—
(RE)-DS-U-ResnetFCN34 [22]	0.772	—
SegNet* [1]	0.767	—
Challenge winners [37]	0.765	—
Tiramisu* [18]	0.765	—
U-Net* [29]	0.740	—

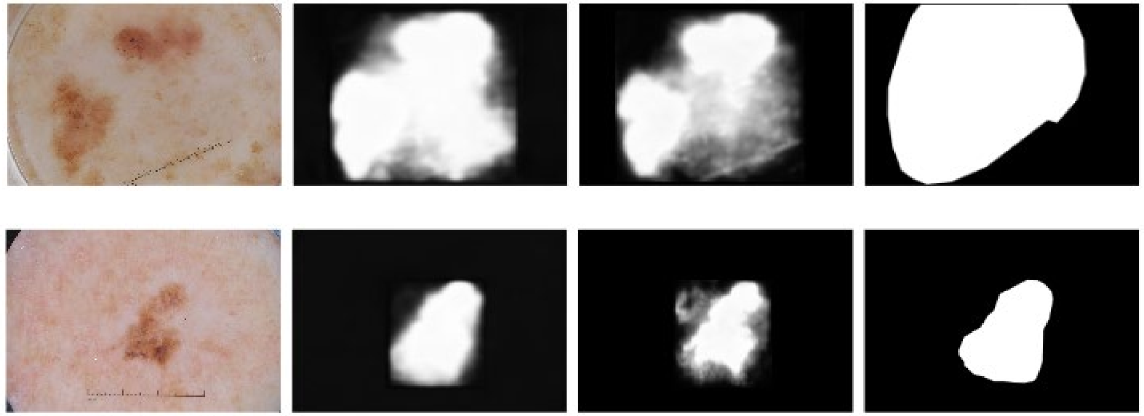






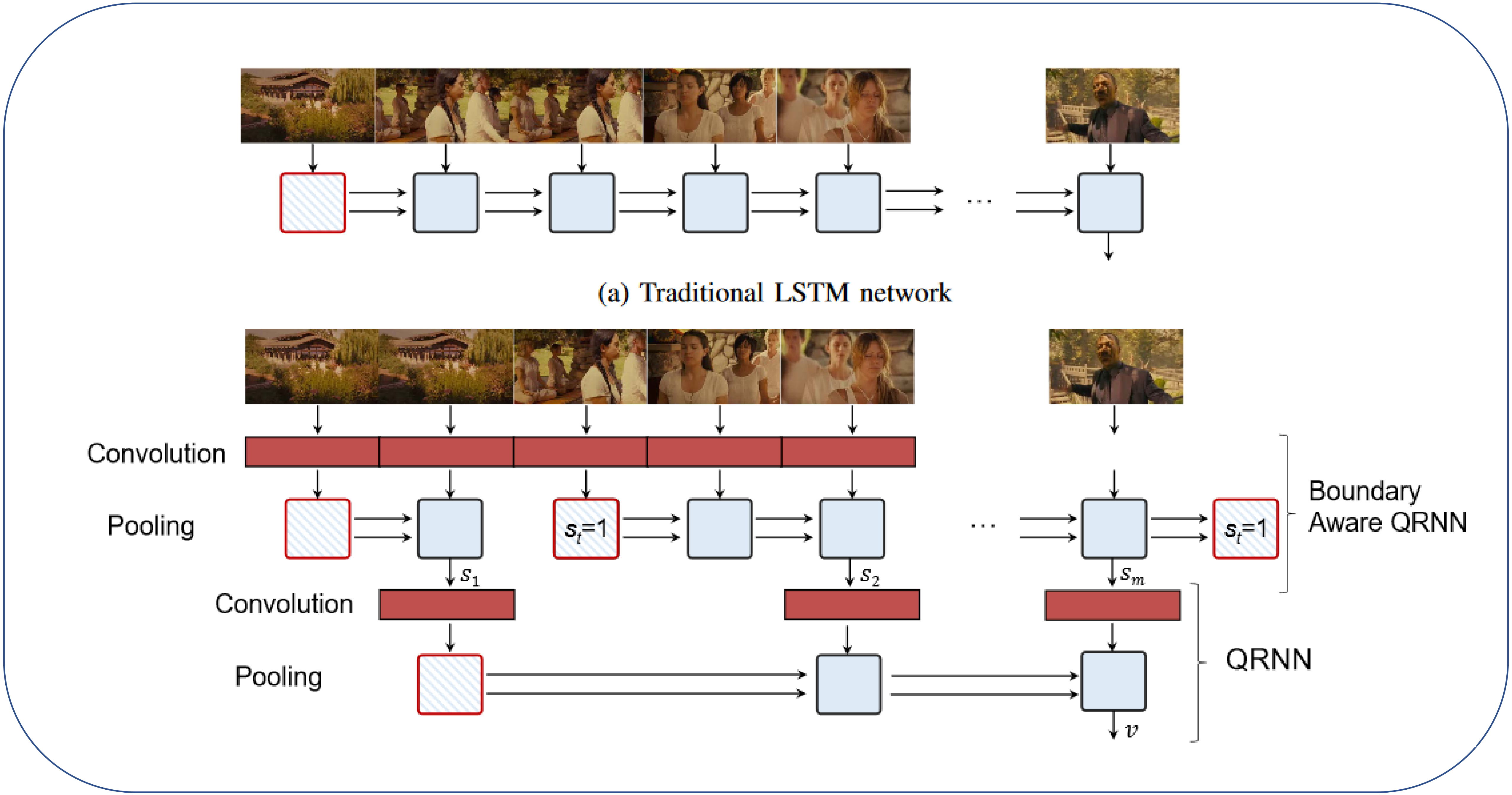
Method	Test IoU	Test TIoU
Ours (ensemble)	**0.850**	0.827
SegAN [35]	0.785	—
GAN Augmented [27]	0.781	—
DCL-PSI [2]	0.777	—
DeepLabv3+* [7]	0.769	—
(RE)-DS-U-ResnetFCN34 [22]	0.772	—
SegNet* [1]	0.767	—
Challenge winners [37]	0.765	—
Tiramisu* [18]	0.765	—
U-Net* [29]	0.740	—



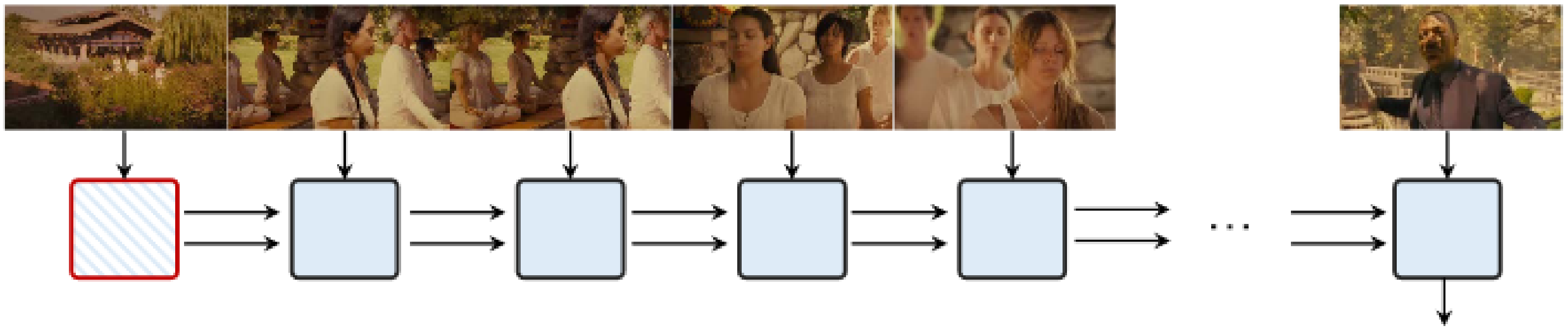


Reproducible Label

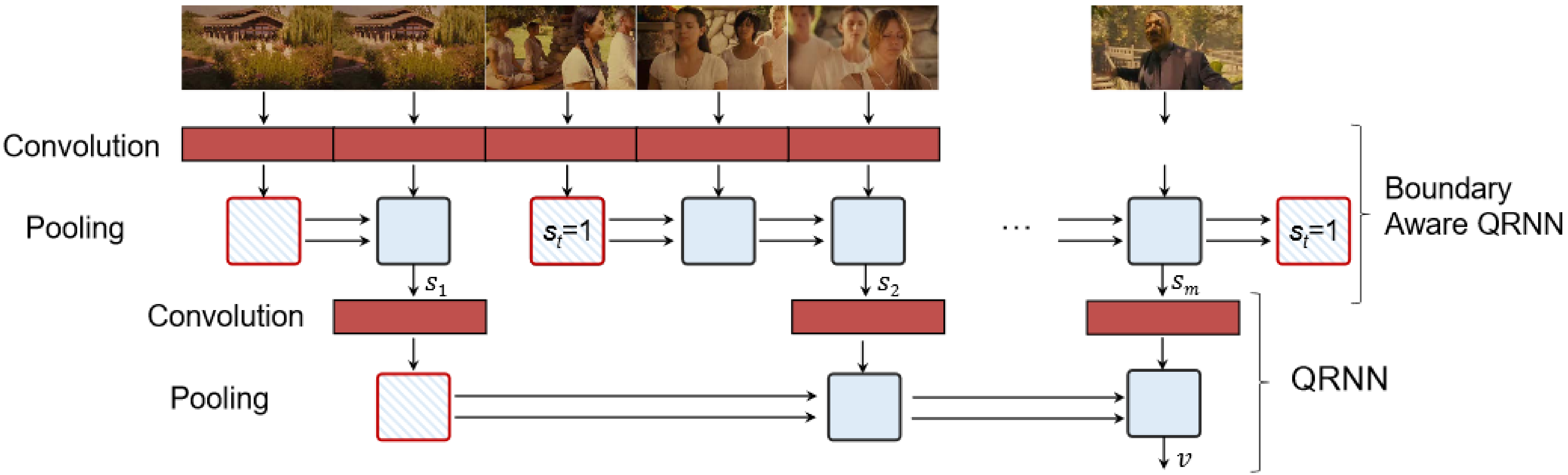
Reproducible Label

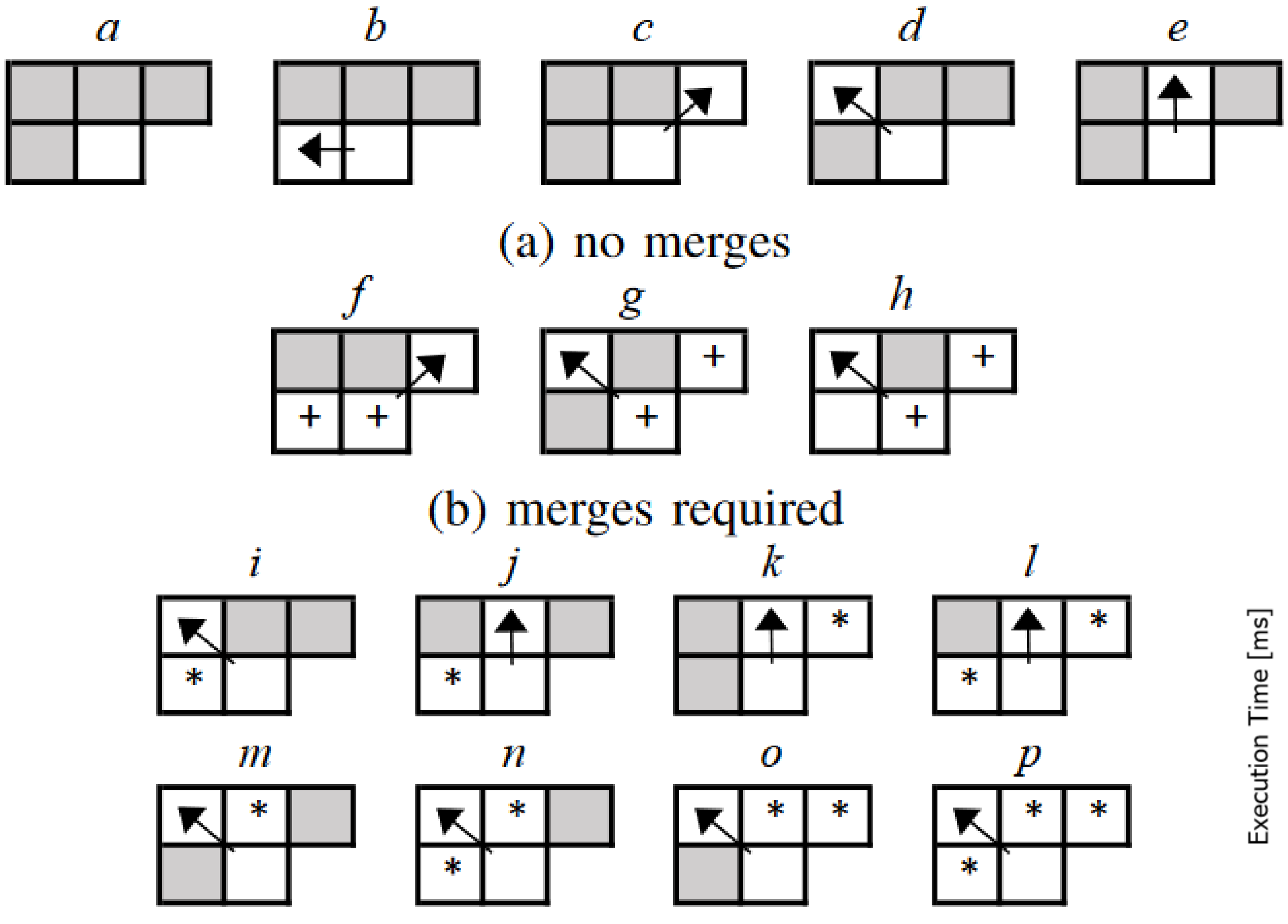


(a) Traditional LSTM network
Convolution
Pooling
s_t=1
s_t=1
Boundary Aware QRNN
s_1
s_2
s_m
Convolution
Pooling
QRNN
v

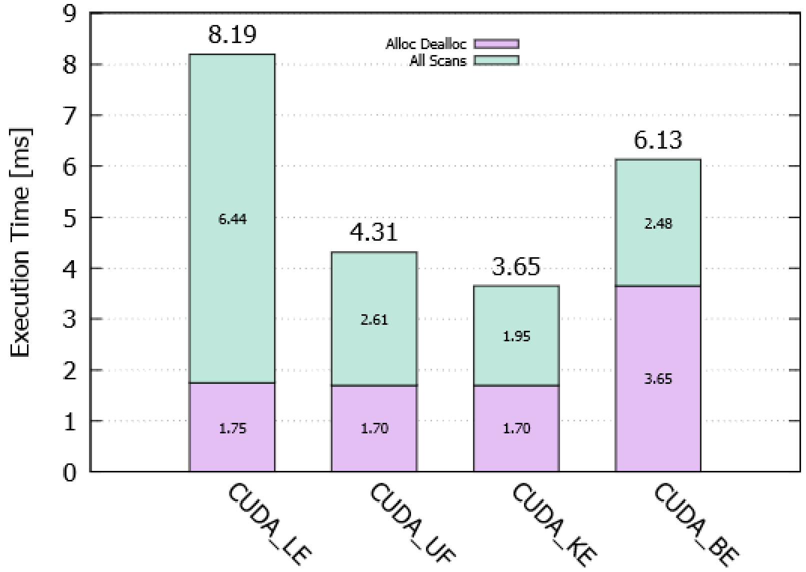


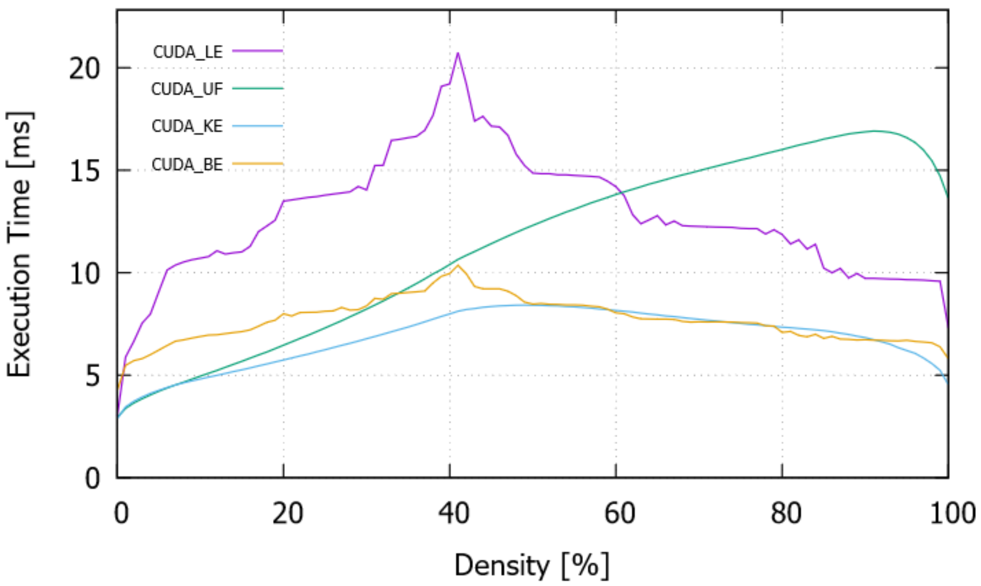
(a) Traditional LSTM network

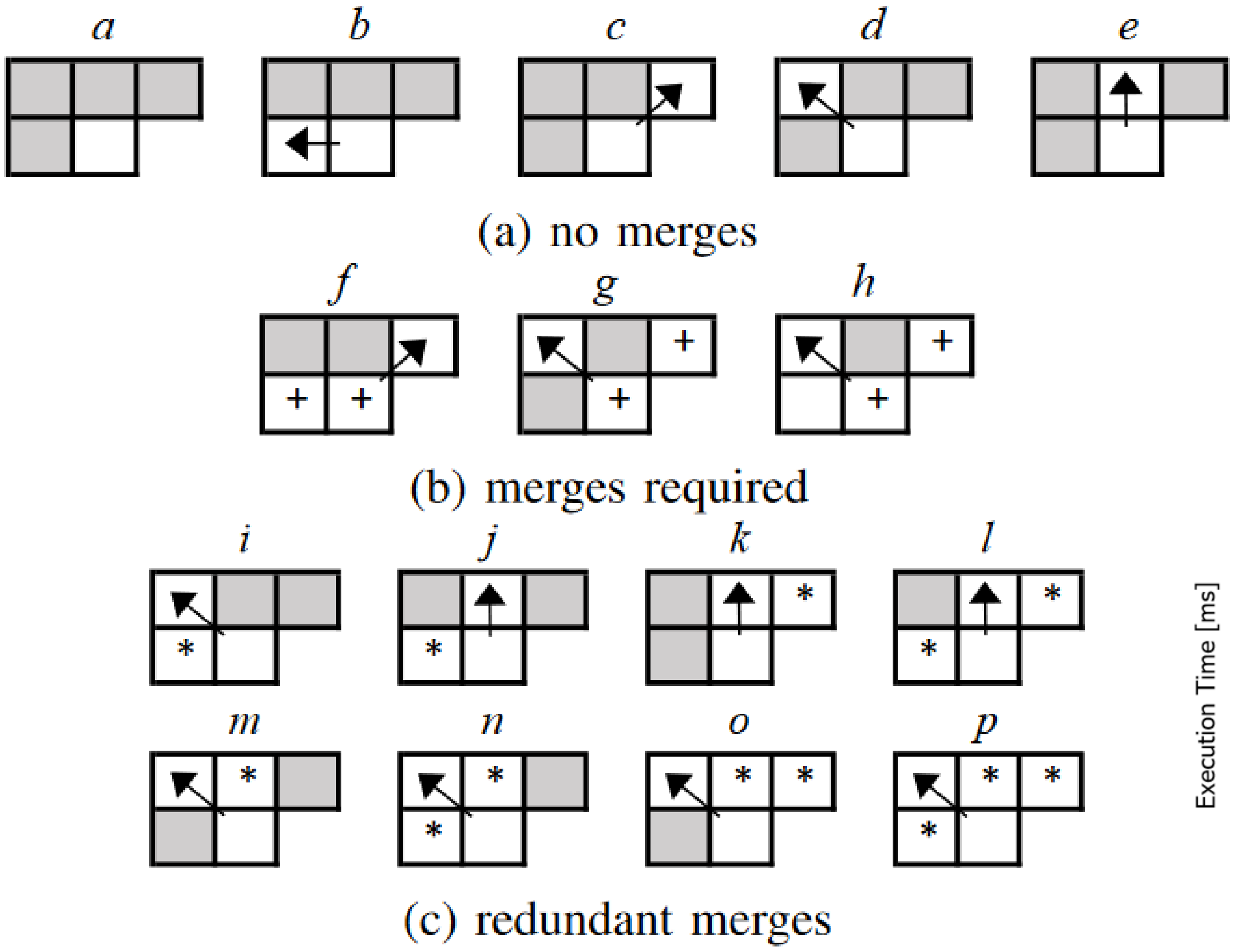


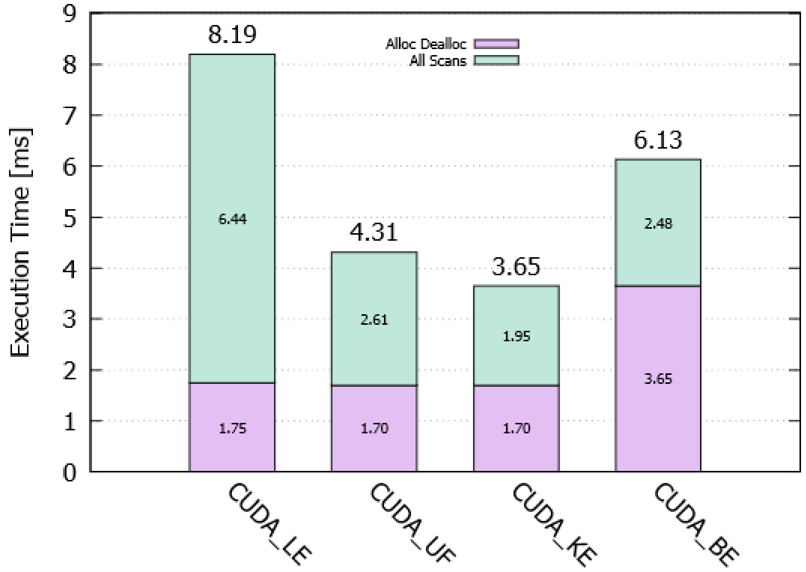


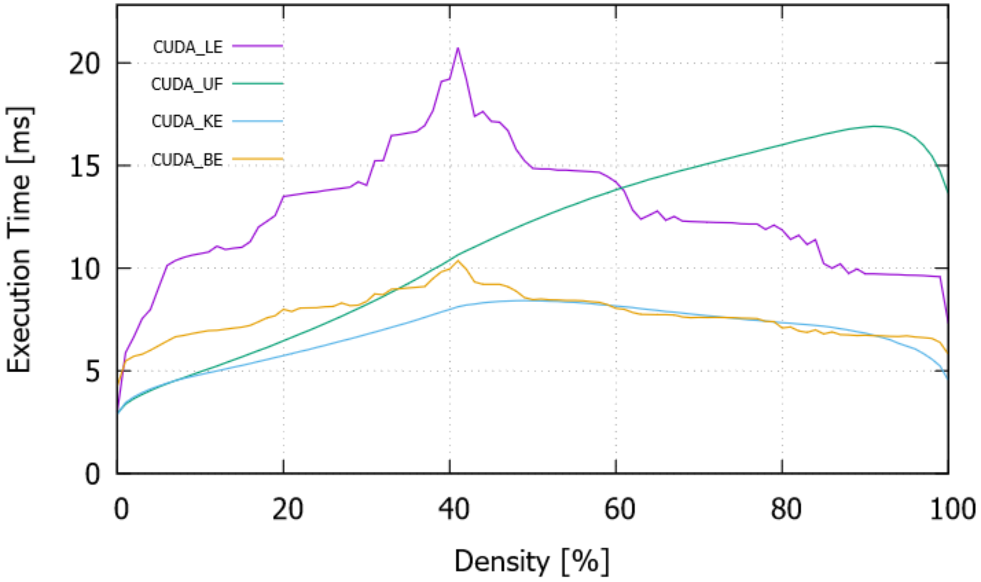
(a) no merges
(b) merges required
(c) redundant merges

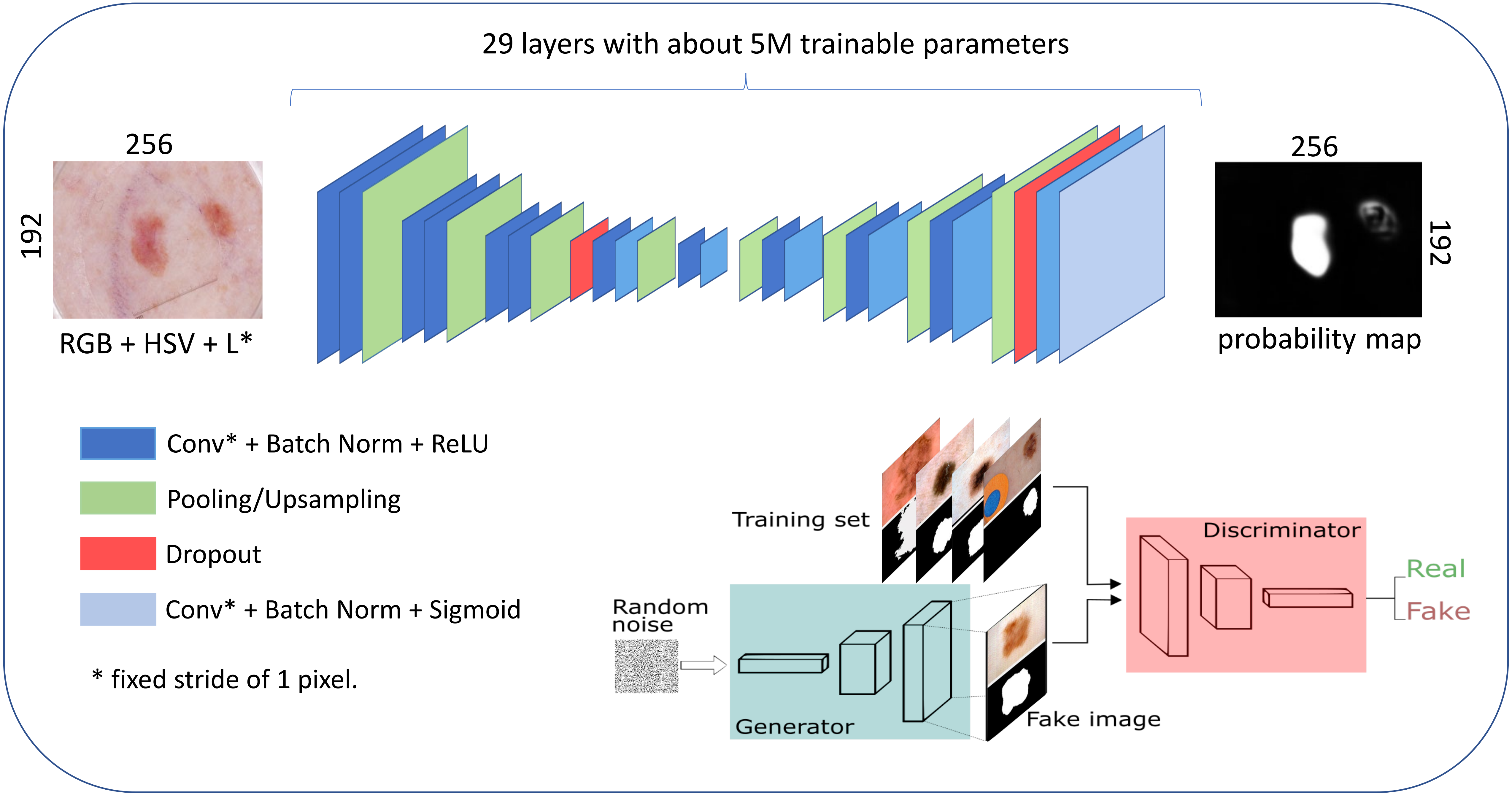




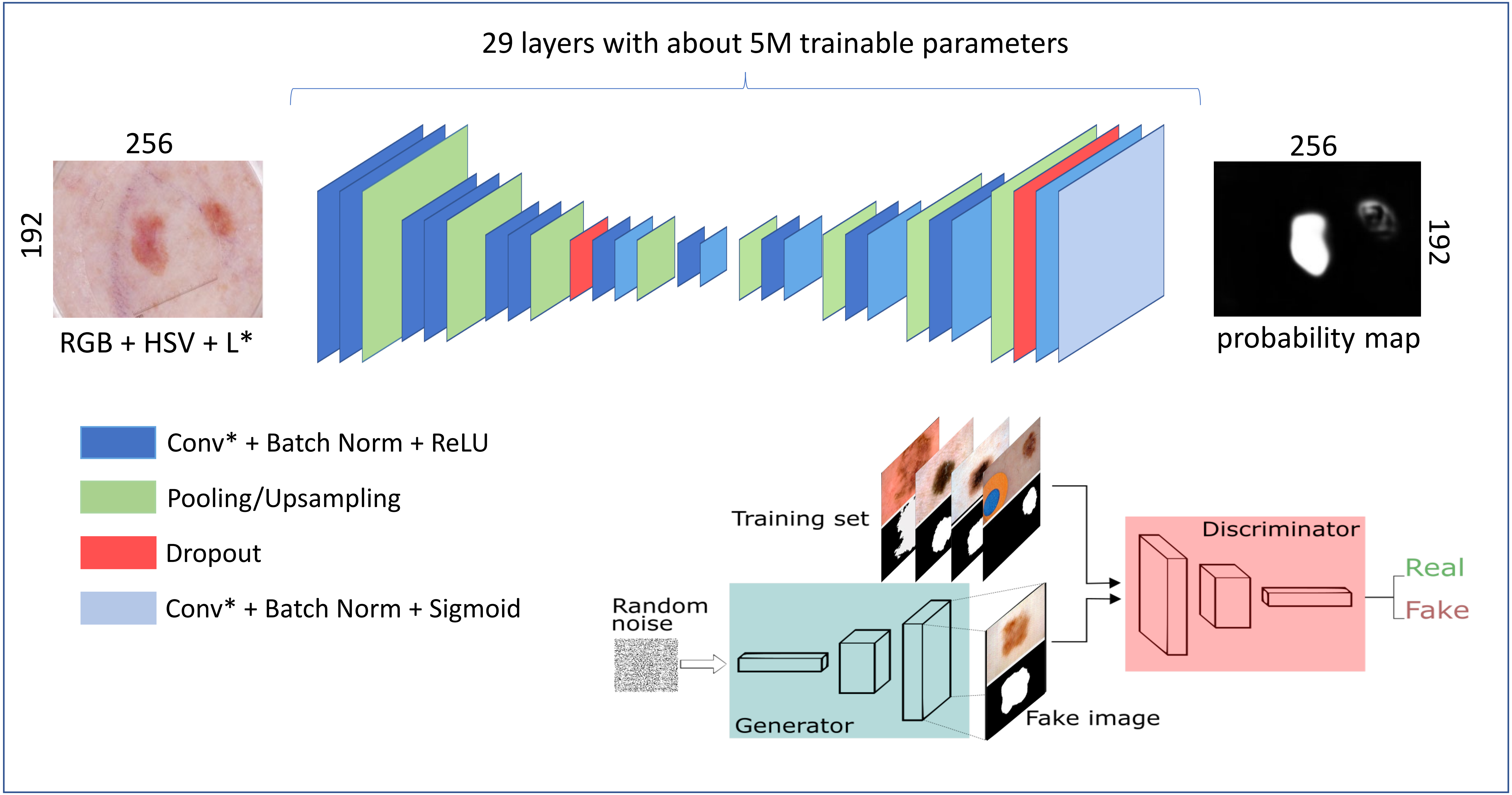

a
b
c
d
e
(a) no merges
f
g
h
(b) merges required
i
j
k
l
m
n
o
p
(c) redundant merges


Alloc Dealloc
All Scans
Execution Time [ms]
8.19
6.44
1.75
4.31
2.61
1.70
3.65
1.95
1.70
6.13
2.48
3.65
CUDA_LE
CUDA_UF
CUDA_KE
CUDA_BE


CUDA_LE
CUDA_UF
CUDA_KE
CUDA_BE
Execution Time [ms]
Density [%]

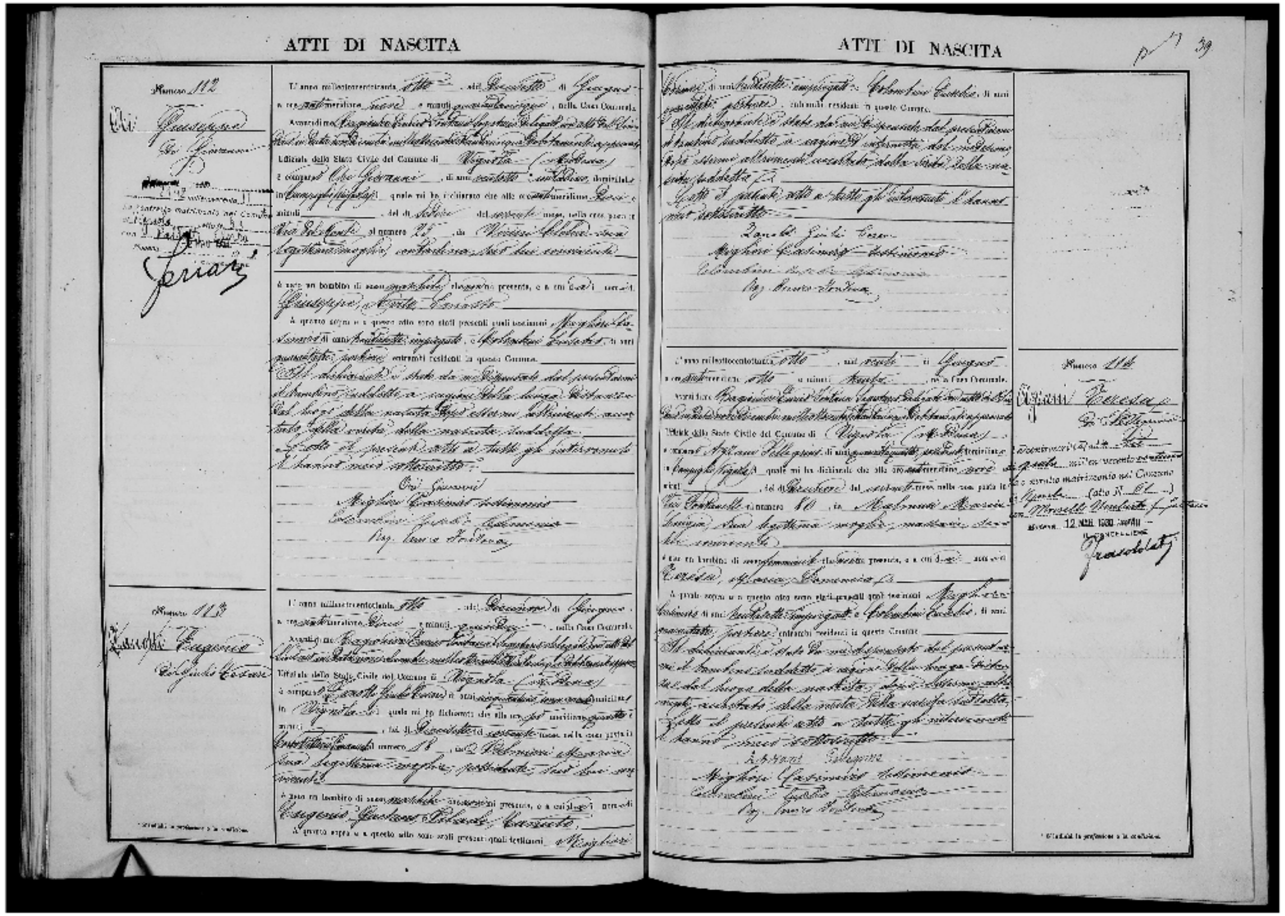


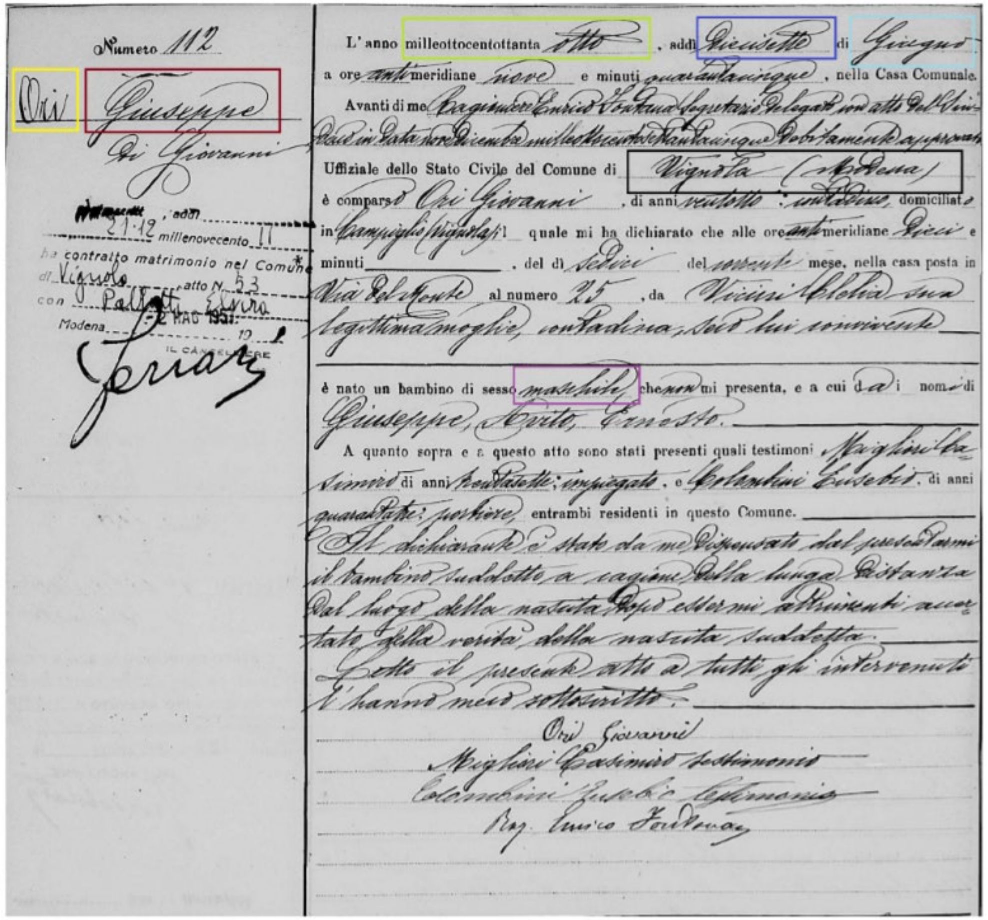
29 layers with about 5M trainable parameters
256
192
RGB + HSV + L*
256
192
probability map
Conv* + Batch Norm + ReLU
Pooling/Upsampling
Dropout
Conv* + Batch Norm + Sigmoid
* fixed stride of 1 pixel.
Training set
Random noise
Generator
Fake image
Discriminator
Real
Fake

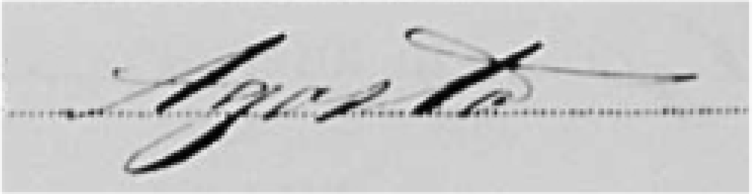

29 layers with about 5M trainable parameters
256
192
RGB + HSV + L*
256
192
probability map
Conv* + Batch Norm + ReLU
Pooling/Upsampling
Dropout
Conv* + Batch Norm + Sigmoid
Training set
Discriminator
Real
Fake
Random noise
Generator
Fake image

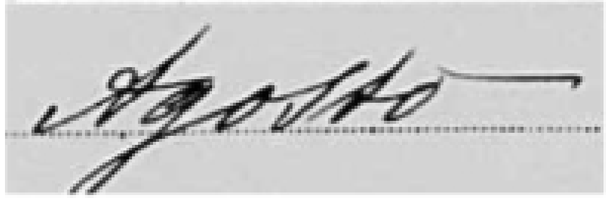
							assign				merge	
x	p	q	r	s	no action	new label	x = p	x = q	x = r	x = s	x = p + r	x = r + s
0	-	-	-	-	**1**							
1	0	0	0	0		**1**						
1	1	0	0	0			**1**					
1	0	1	0	0				**1**				
1	0	0	1	0					**1**			
1	0	0	0	1						**1**		
1	1	1	0	0			*1*	**1**				
1	1	0	1	0							**1**	
1	1	0	0	1			*1*			**1**		
1	0	1	1	0				**1**	*1*			
1	0	1	0	1				**1**		*1*		
1	0	0	1	1								**1**
1	1	1	1	0			*1*	**1**	*1*			
1	1	1	0	1			*1*	**1**		*1*		
1	1	0	1	1							*1*	**1**
1	0	1	1	1				**1**	*1*	*1*		
1	1	1	1	1			*1*	**1**	*1*	*1*		

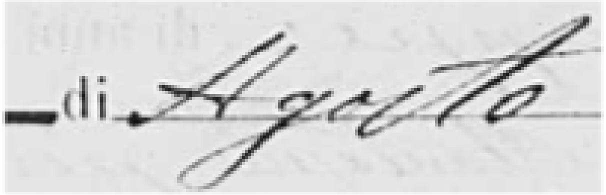
							assign				merge	
x	p	q	r	s	no action	new label	x = p	x = q	x = r	x = s	x = p + r	x = r + s
0	-	-	-	-	**1**							
1	0	0	0	0		**1**						
1	1	0	0	0			**1**					
1	0	1	0	0				**1**				
1	0	0	1	0					**1**			
1	0	0	0	1						**1**		
1	1	1	0	0			*1*	**1**				
1	1	0	1	0							**1**	
1	1	0	0	1			*1*			**1**		
1	0	1	1	0				**1**	*1*			
1	0	1	0	1				**1**		*1*		
1	0	0	1	1								**1**
1	1	1	1	0			*1*	**1**	*1*			
1	1	1	0	1			*1*	**1**		*1*		
1	1	0	1	1							*1*	**1**
1	0	1	1	1				**1**	*1*	*1*		
1	1	1	1	1			*1*	**1**	*1*	*1*		

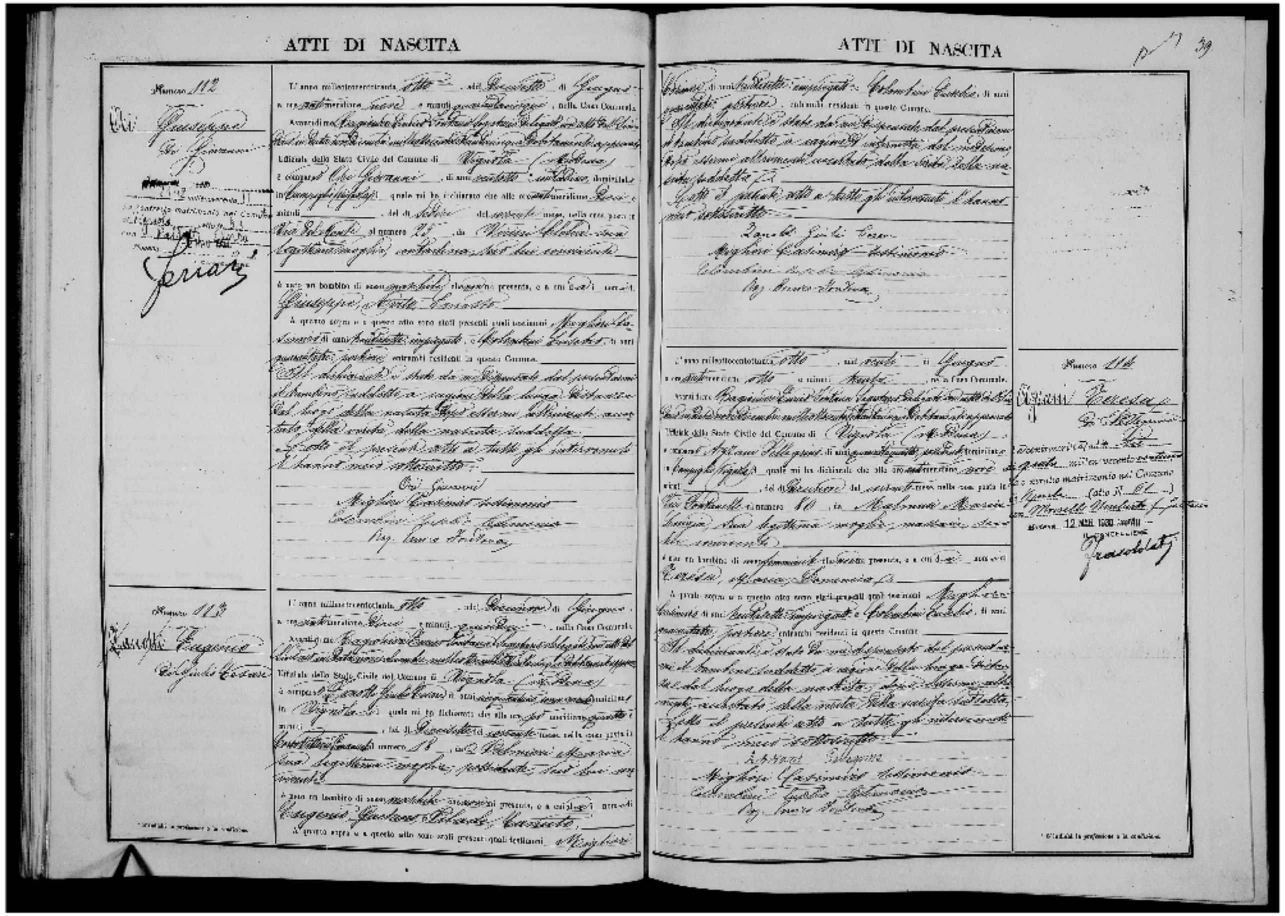

ATTI DI NASCITA
ATTI DI NASCITA


Numero 112
Ori Giuseppe
di Giovanni
L'anno milleottocentottanta otto, addì diciotto di Giugno
a ore antimeridiane nove e minuti quarantacinque, nella Casa Comunale.
Avanti di me
Uffiziale dello Stato Civile del Comune di Vignola (Modena)
è comparso Ori Giovanni, di anni ventotto
quale mi ha dichiarato che alle ore antimeridiane dieci e minuti
del dì sedici del corrente mese, nella casa posta in Via del Monte al numero 25, da
legittima moglie, contadina, seco lui convivente
è nato un bambino di sesso maschile, che non mi presenta, e a cui dà i nomi di
Giuseppe, Agosto, Ernesto.
A quanto sopra e a questo atto sono stati presenti quali testimoni
di anni
entrambi residenti in questo Comune.
Ori Giovanni

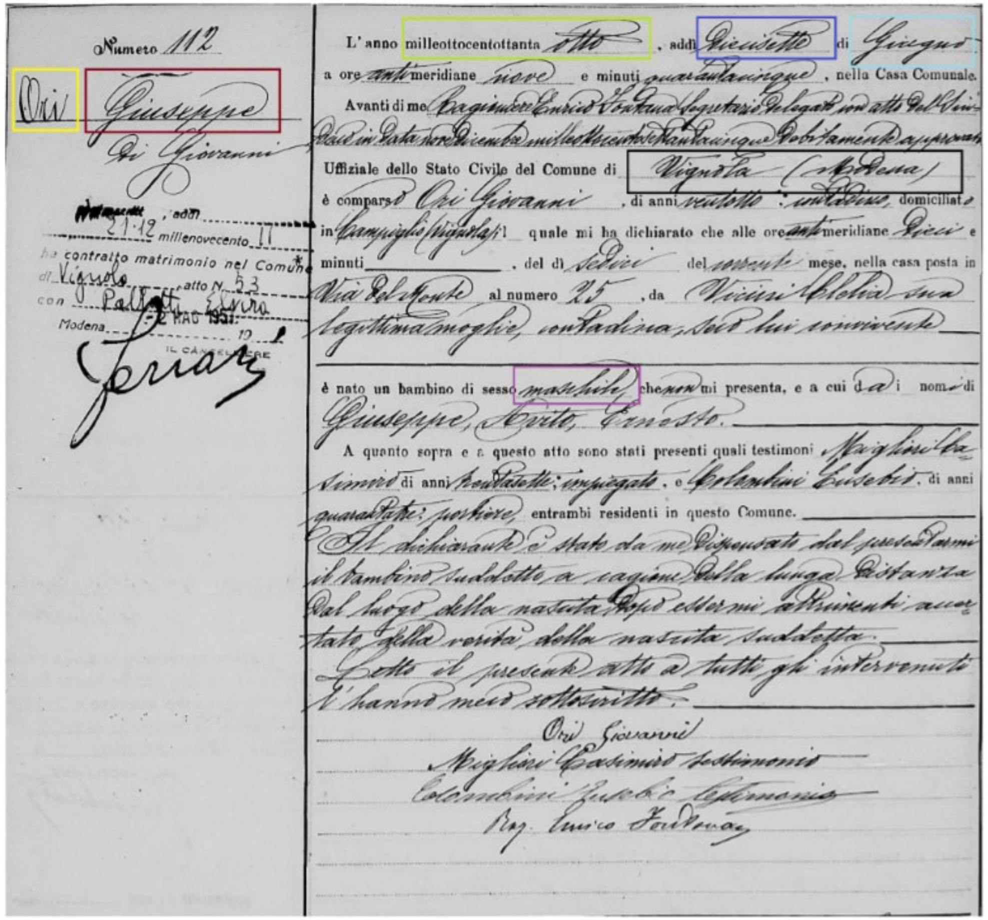

Agosto


Agosto


di Agosto


ATTI DI NASCITA

ATTI DI NASCITA


Numero 112

Ori Giuseppe di Giovanni

Annotato 21-12 addì [illegible] millenovecento 11 ha contratto matrimonio nel Comune di Vignola atto N. 53 con Pallotti Elvira
Modena 2 MAG 1951
IL CANCELLIERE

L'anno milleottocentottanta otto, addì diciotto di Giugno a ore anti meridiane nove e minuti quarantacinque, nella Casa Comunale.

Avanti di me Ragioniere Enrico Fontana Segretario Delegato con atto del [illegible] in data [illegible]

Ufficiale dello Stato Civile del Comune di Vignola (Modena)

è comparso Ori Giovanni, di anni ventotto, [illegible], domiciliato in Campiglio (Vignola), quale mi ha dichiarato che alle ore anti meridiane dieci e minuti [illegible], del dì sedici del corrente mese, nella casa posta in Via del Monte al numero 25, da Vaccari Adelia sua legittima moglie, contadina, seco lui convivente

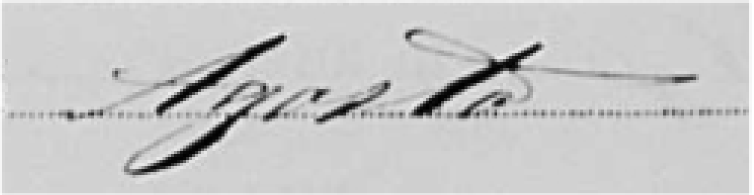
è nato un bambino di sesso maschile, che non mi presenta, e a cui dà i nomi di Giuseppe, Agosto, Ernesto.

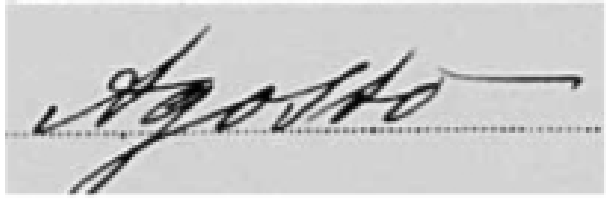
A quanto sopra e a questo atto sono stati presenti quali testimoni Migliori Casimiro di anni [illegible], impiegato, e Colombini Eusebio di anni quaranta[illegible], portiere, entrambi residenti in questo Comune.

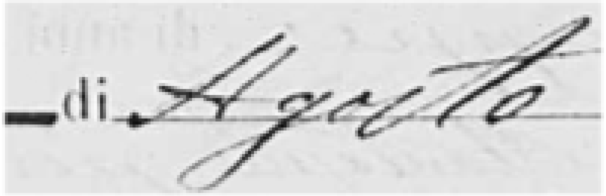
Il dichiarante è stato da me dispensato dal presentarmi il bambino suddetto a cagione della lunga distanza dal luogo della nascita, dopo essermi altrimenti accertato della verità della nascita suddetta.

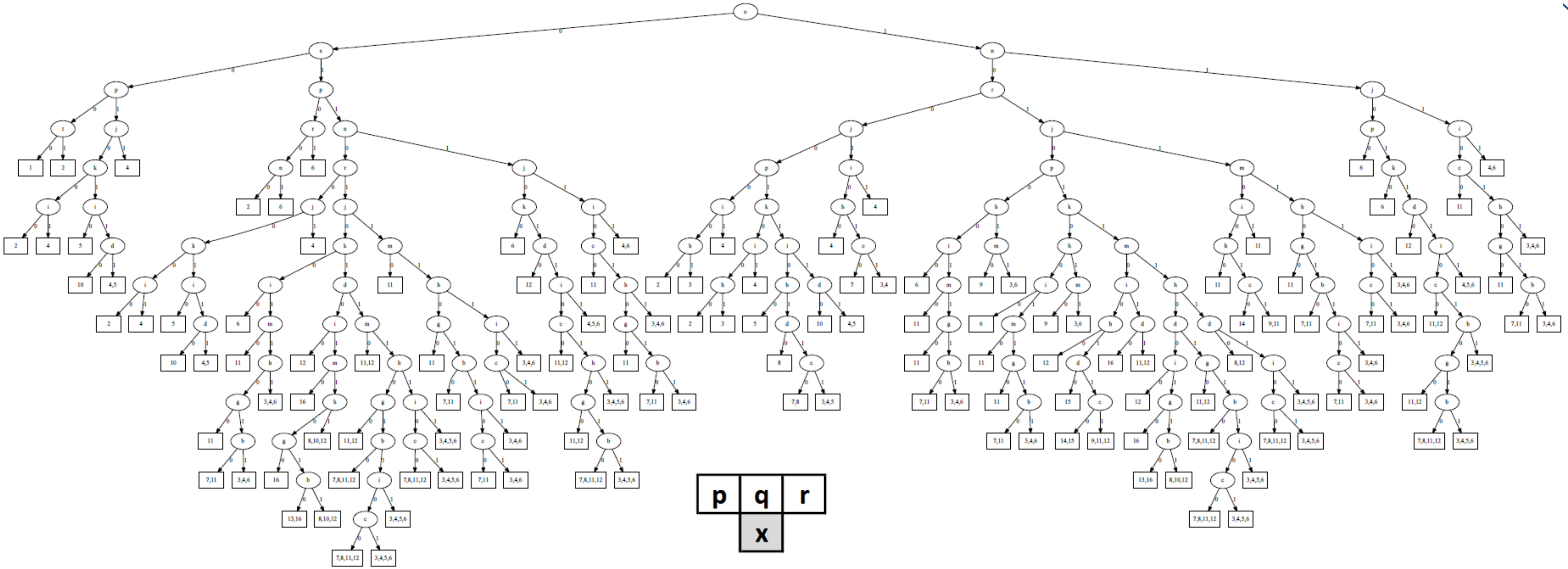
Letto il presente atto a tutti gli intervenuti l'hanno meco sottoscritto.

Ori Giovanni
Migliori Casimiro testimonio
Colombini Eusebio testimonio
Rag. Enrico Fontana







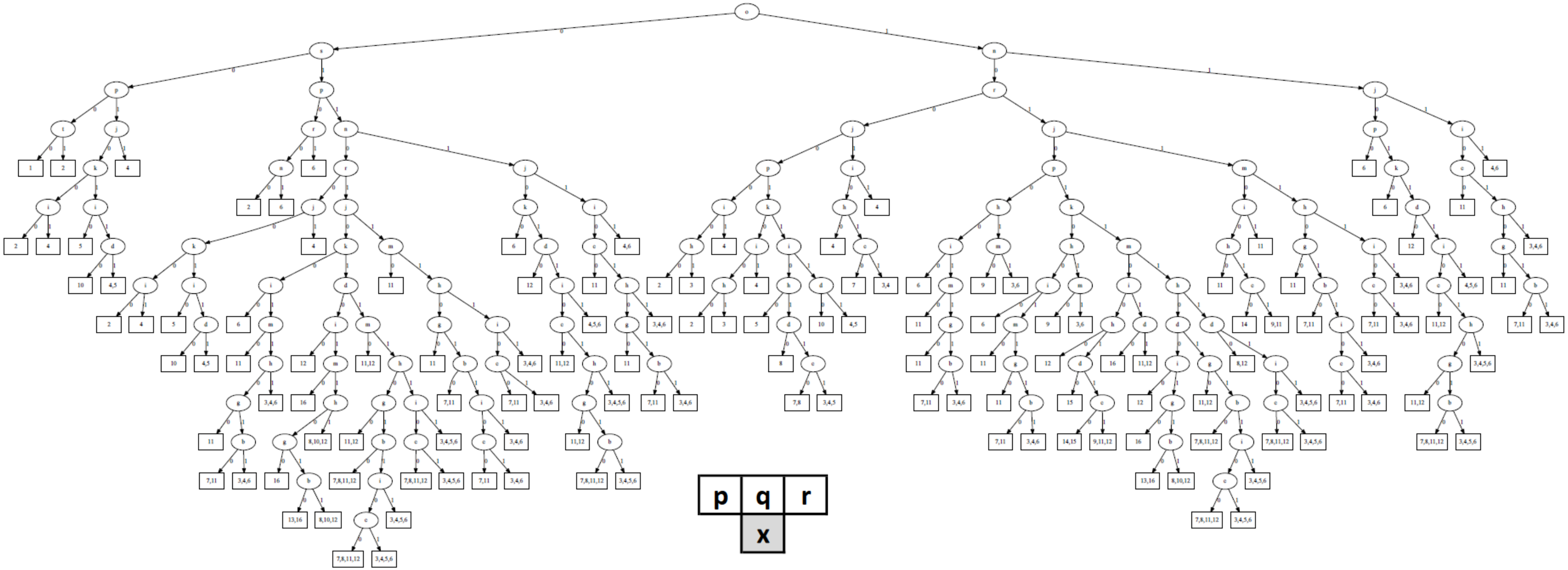


p	q	r
	x	

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
0	1	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	2	2	2	0	0	0	3	0	4	0	0	5	5	0	0
0	0	34	0	0	0	35	0	0	0	36	0	37	37	37	0	0	38	0	0	39	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	38	38	39	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	0	0	0	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	0	0	40	0
0	34	34	34	0	0	0	35	35	35	0	0	37	37	37	0	0	0	38	38	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

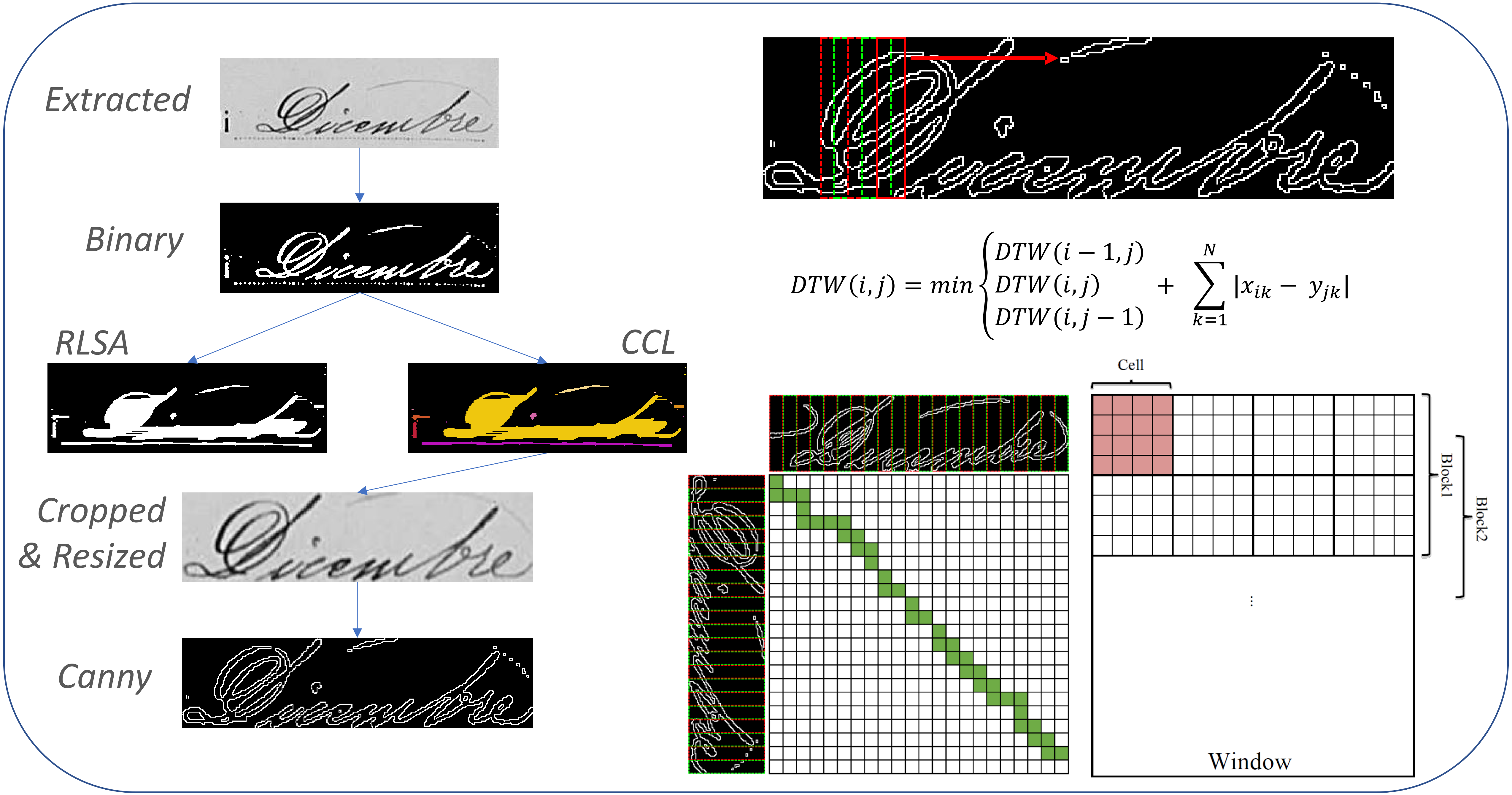
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	2	2	2	0	0	0	3	0	3	0	0	4	4	0	0
0	0	1	0	0	0	2	0	0	0	2	0	3	3	3	0	0	4	0	0	4	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	4	4	4	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	0	0	0	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	0	0	4	0
0	1	1	1	0	0	0	2	2	2	0	0	3	3	3	0	0	0	4	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



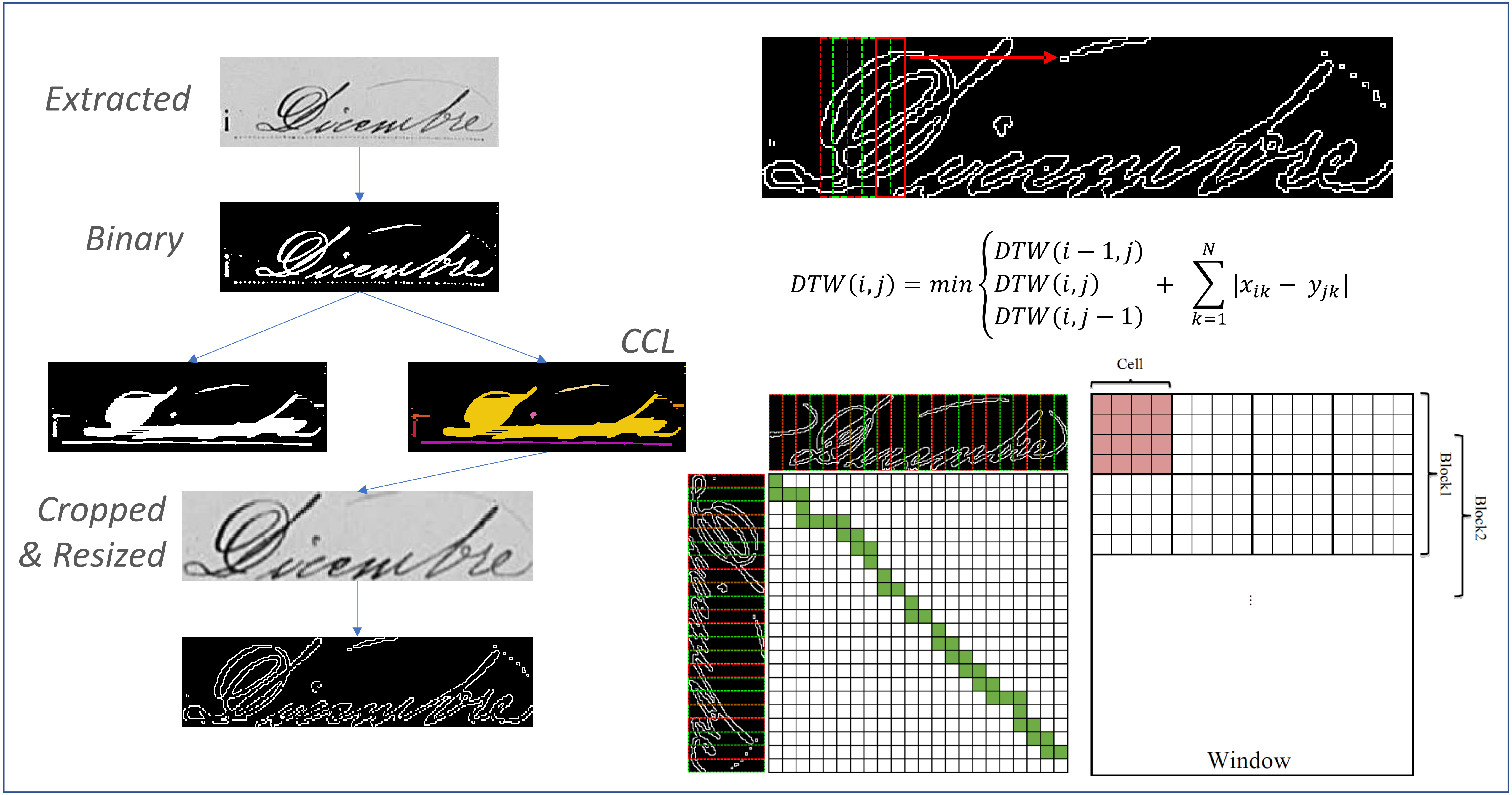
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
0	1	1	1	0	0	0	1	1	1	0	0	1	1	1	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	2	2	2	0	0	0	3	0	4	0	0	5	5	0	0
0	0	34	0	0	0	35	0	0	0	36	0	37	37	37	0	0	38	0	0	39	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	38	38	39	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	0	0	0	0
0	0	34	0	0	0	35	0	0	0	36	0	0	37	0	0	0	38	0	0	40	0
0	34	34	34	0	0	0	35	35	35	0	0	37	37	37	0	0	0	38	38	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

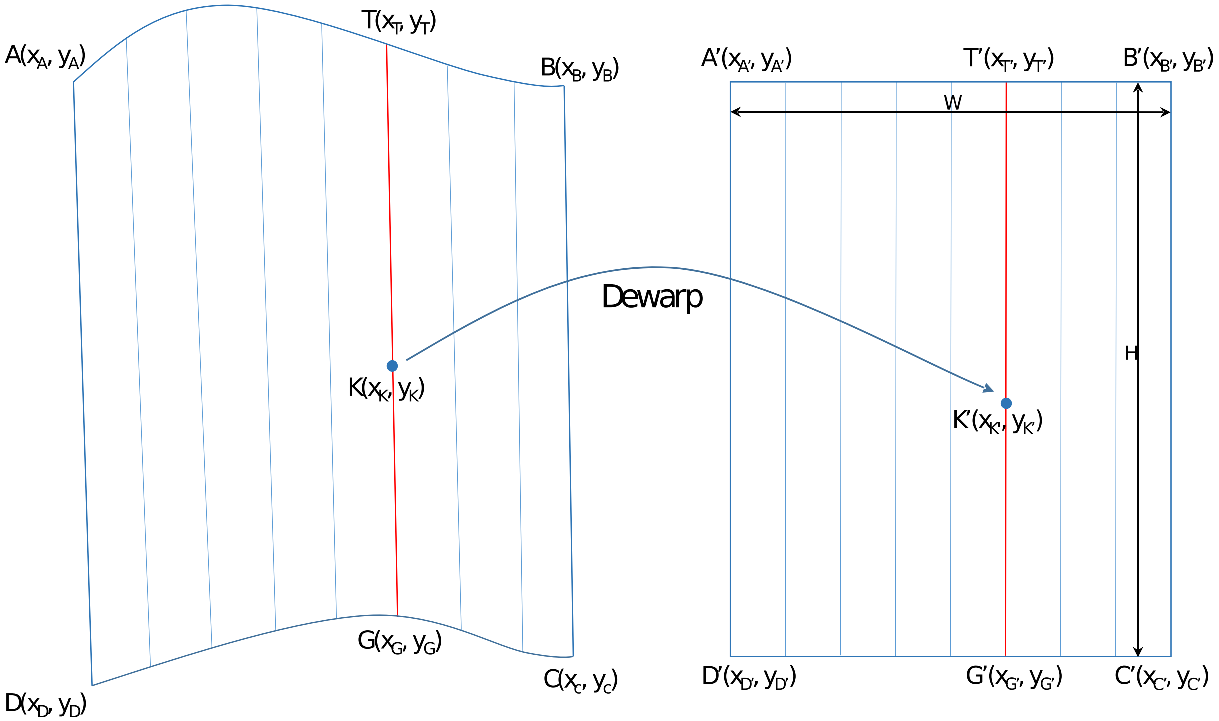
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	2	2	2	0	0	0	3	0	3	0	0	4	4	0	0
0	0	1	0	0	0	2	0	0	0	2	0	3	3	3	0	0	4	0	0	4	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	4	4	4	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	0	0	0	0
0	0	1	0	0	0	2	0	0	0	2	0	0	3	0	0	0	4	0	0	4	0
0	1	1	1	0	0	0	2	2	2	0	0	3	3	3	0	0	0	4	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



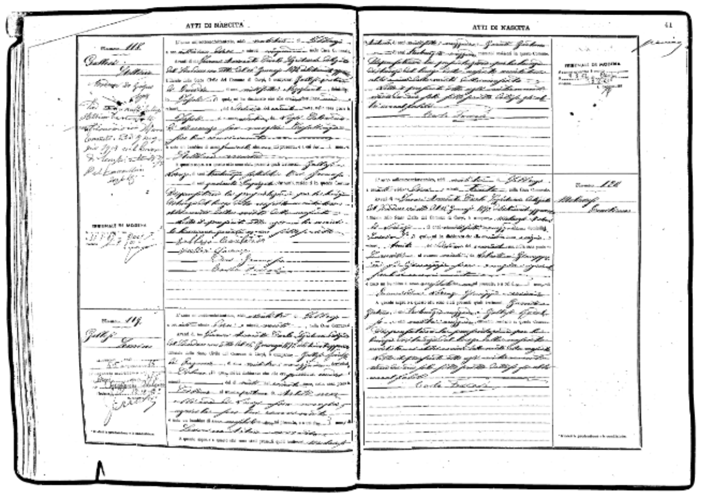
Extracted
Binary
RLSA
CCL
Cropped & Resized
Canny
$DTW(i,j) = min\begin{cases} DTW(i-1,j) \\ DTW(i,j) \\ DTW(i,j-1) \end{cases} + \sum_{k=1}^{N} |x_{ik} - y_{jk}|$
Cell
Block1
Block2
Window

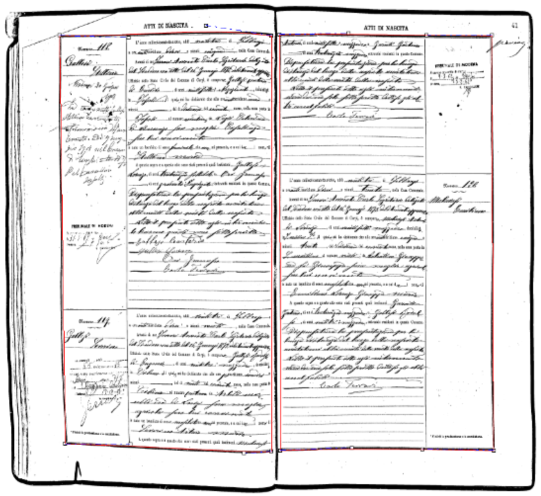


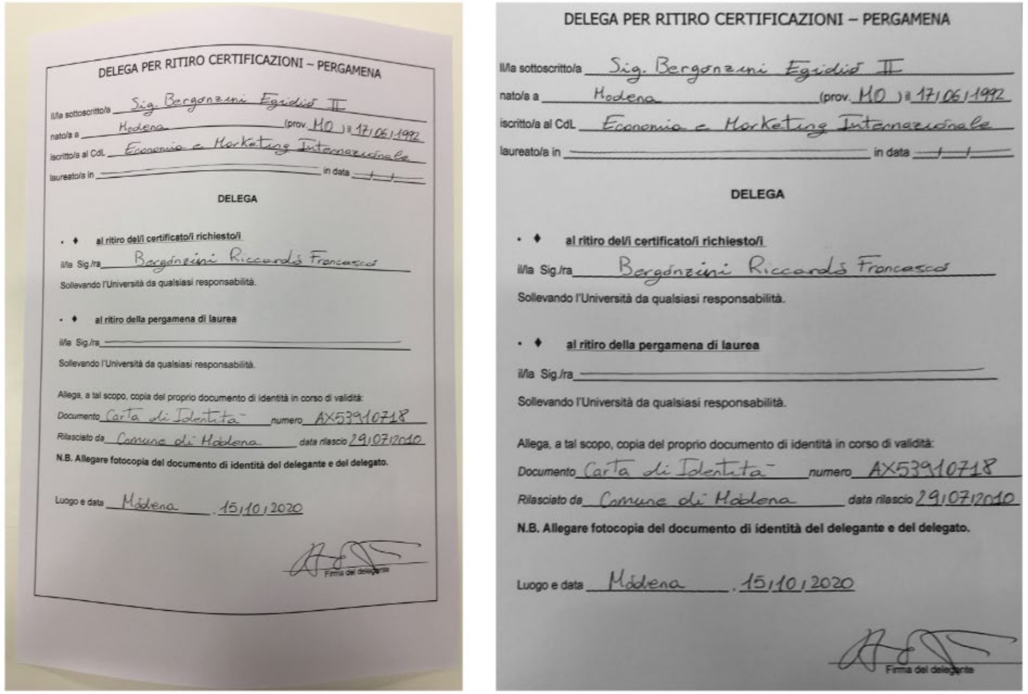
Extracted
Binary
CCL
Cropped
& Resized
$DTW(i,j) = min \begin{cases} DTW(i-1,j) \\ DTW(i,j) \\ DTW(i,j-1) \end{cases} + \sum_{k=1}^{N} |x_{ik} - y_{jk}|$
Cell
Block1
Block2
Window



$$
\begin{cases}
x'_k = x'_A + W * \dfrac{|\widehat{AT}|}{|\widehat{AB}|} \\
y'_k = y'_A + H * \dfrac{|TK|}{|TG|}
\end{cases}
$$







DELEGA PER RITIRO CERTIFICAZIONI – PERGAMENA

Il/la sottoscritto/a Sig. Bergonzini Egidio II

nato/a a Modena (prov. MO) il 17/06/1992

iscritto/a al CdL Economia e Marketing Internazionale

laureato/a in ______ in data __/__/__

DELEGA

- al ritiro del/i certificato/i richiesto/i

Il/la Sig./ra Bergonzini Riccardo Francesco

Sollevando l'Università da qualsiasi responsabilità.

- al ritiro della pergamena di laurea

Il/la Sig./ra ______

Sollevando l'Università da qualsiasi responsabilità.

Allega, a tal scopo, copia del proprio documento di identità in corso di validità:

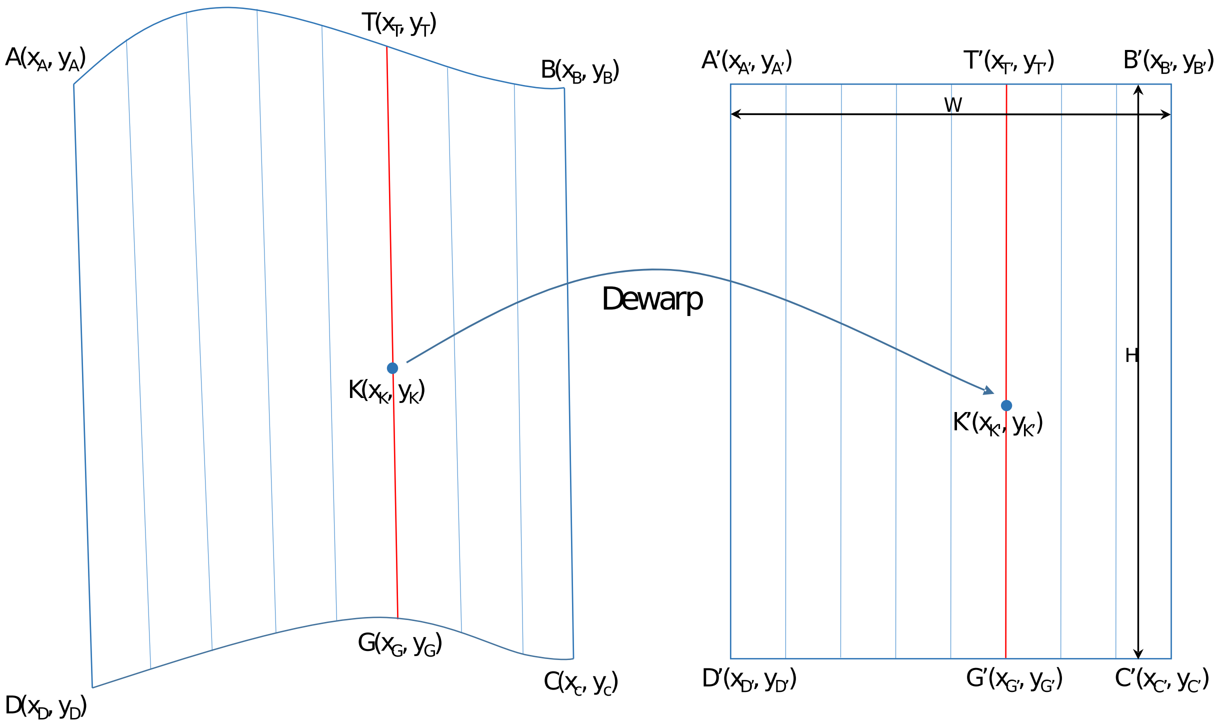
Documento Carta di Identità numero AX53910718

Rilasciato da Comune di Modena data rilascio 29/07/2010

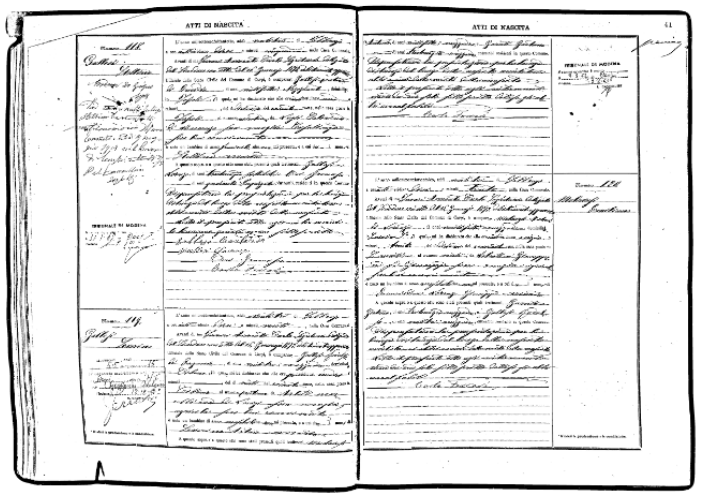
N.B. Allegare fotocopia del documento di identità del delegante e del delegato.

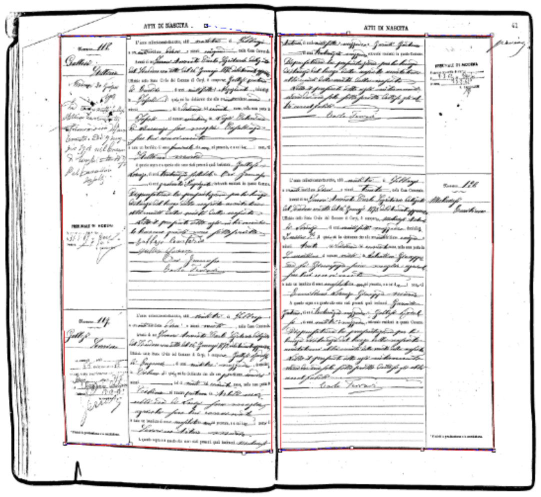
Luogo e data Modena 15/10/2020

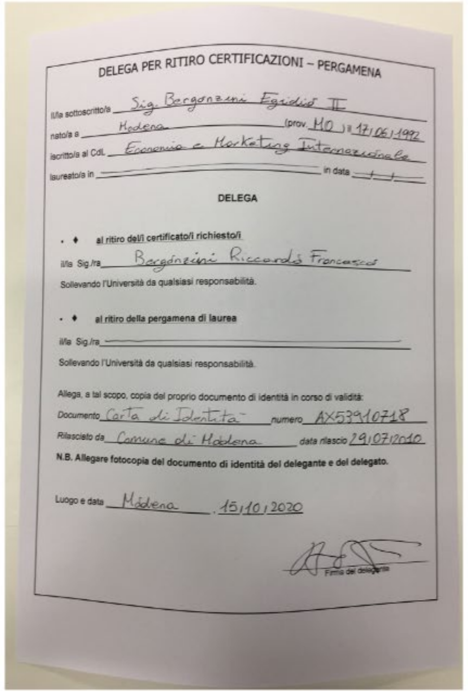
Firma del delegante

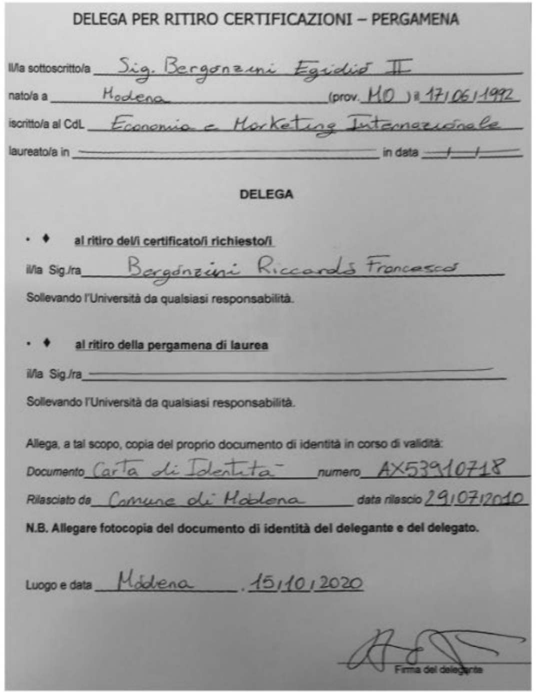


$$
\begin{cases}
x'_k = x'_A + W * \dfrac{|\widehat{AT}|}{|\widehat{AB}|} \\
y'_k = y'_A + H * \dfrac{|TK|}{|TG|}
\end{cases}
$$









DELEGA PER RITIRO CERTIFICAZIONI – PERGAMENA

Il/la sottoscritto/a Sig. Bergonzini Egidio II

nato/a a Modena (prov. MO) il 17/06/1992

iscritto/a al CdL Economia e Marketing Internazionale

laureato/a in in data / /

DELEGA

- al ritiro del/i certificato/i richiesto/i

Il/la Sig./ra Bergonzini Riccardo Francesco

Sollevando l'Università da qualsiasi responsabilità.

- al ritiro della pergamena di laurea

Il/la Sig./ra

Sollevando l'Università da qualsiasi responsabilità.

Allega, a tal scopo, copia del proprio documento di identità in corso di validità:

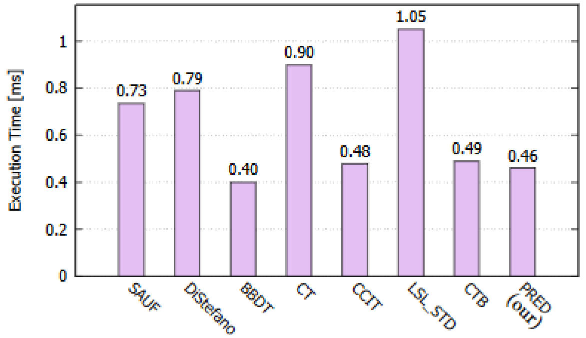
Documento Carta di Identità numero AX53910718

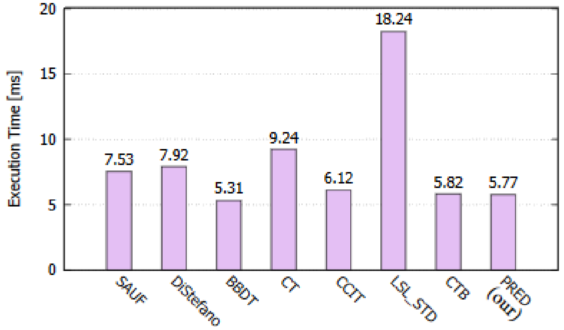
Rilasciato da Comune di Modena data rilascio 29/07/2010

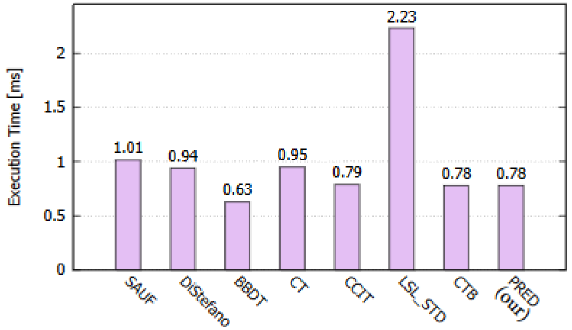
N.B. Allegare fotocopia del documento di identità del delegante e del delegato.

Luogo e data Modena 15/10/2020

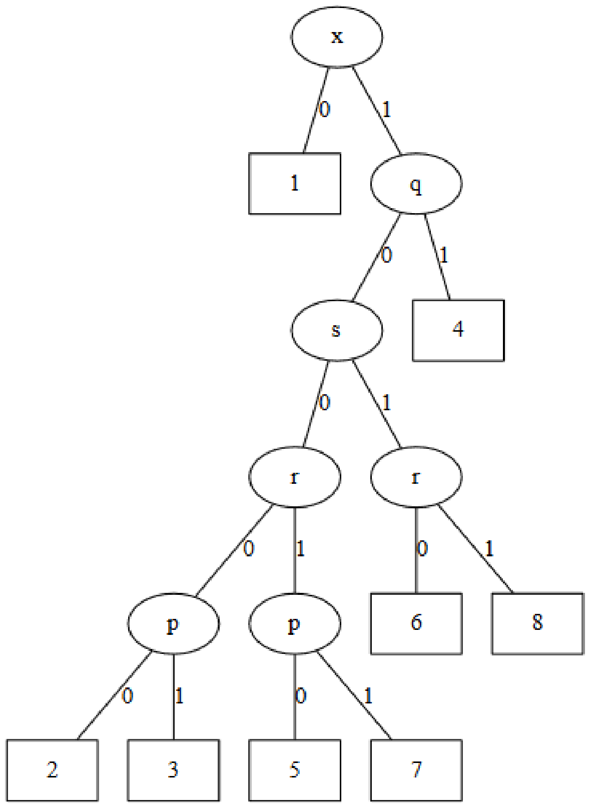
Firma del delegante

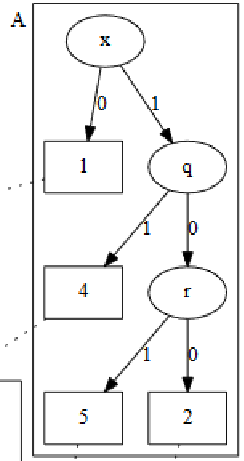


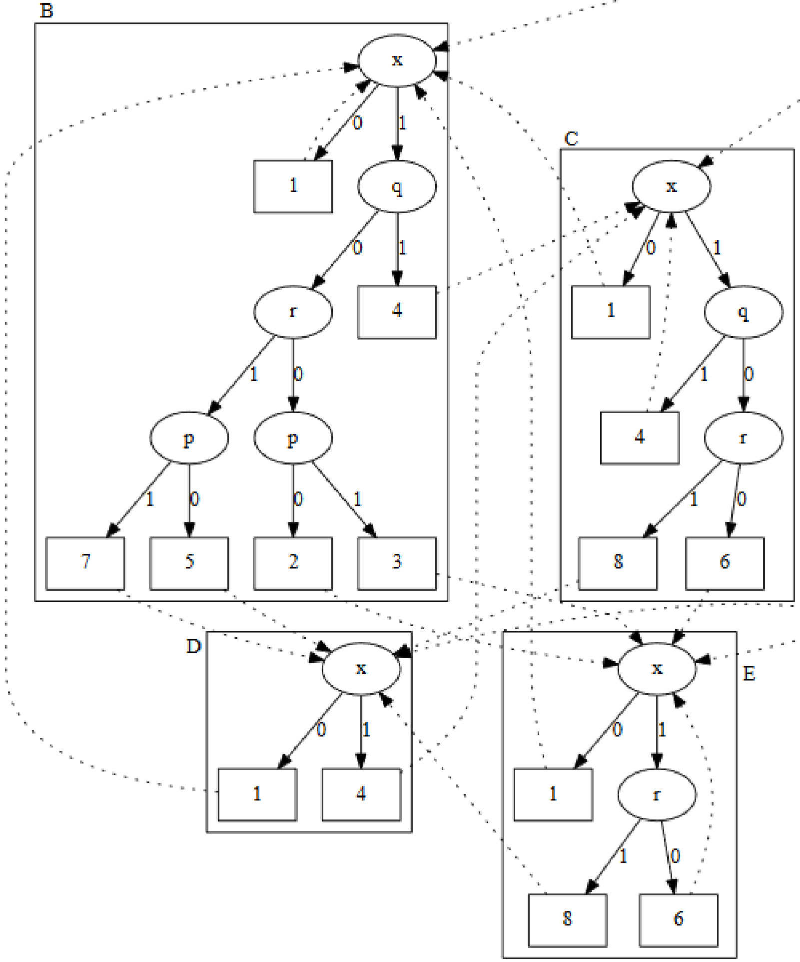


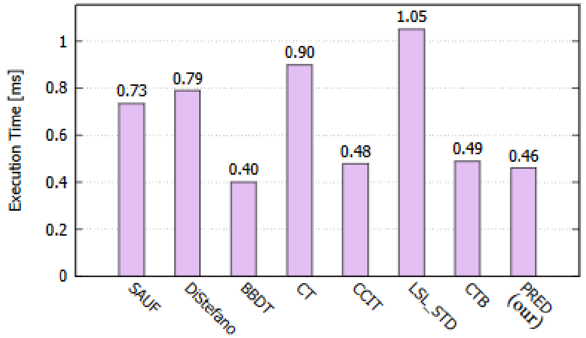


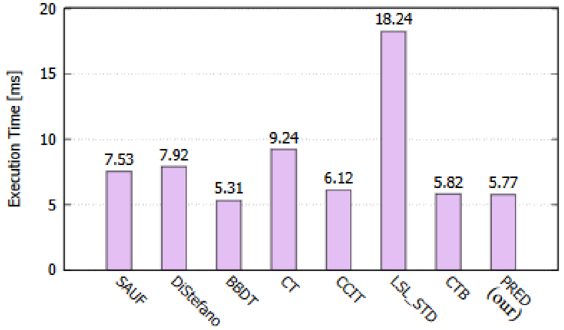

```
...
tree_C:
        if (++c >= w - 1)
                goto break_C;
        if (condition_x) {
                if (condition_q) {
                        // action 4 - x <= q
                        goto tree_C;
                }
                else {
                        if (condition_r) {
                                // action 8 - x <= r + s
                                goto tree_D;
                        }
                        else {
                                // action 6 - x <= s
                                goto tree_E;
                        }
                }
        }
        else {
                // action 1 - do nothing
                goto tree_B;
        }
...
```

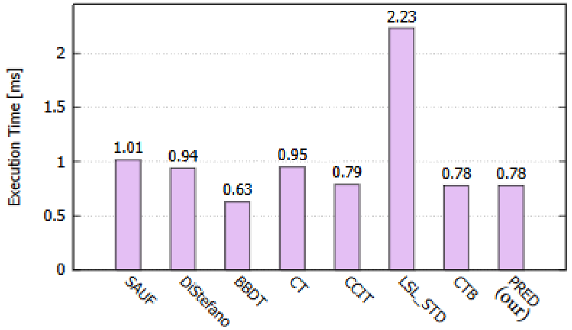




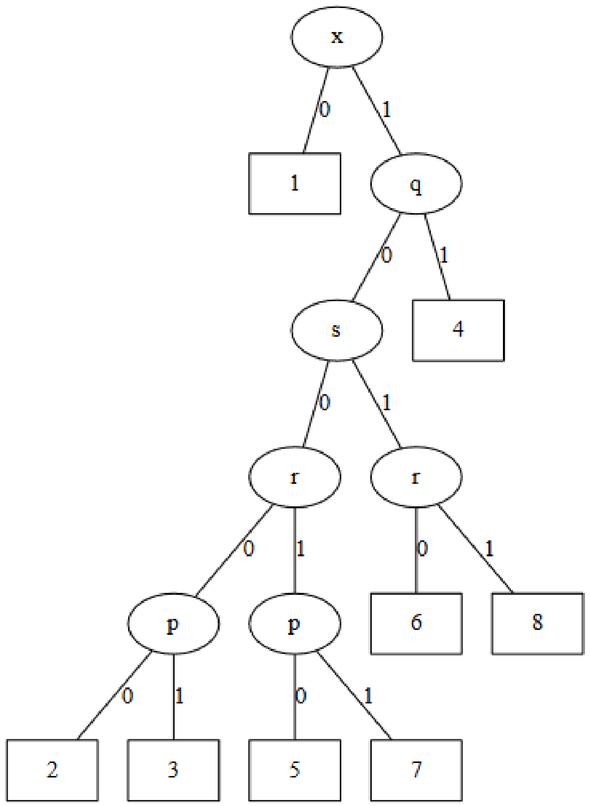


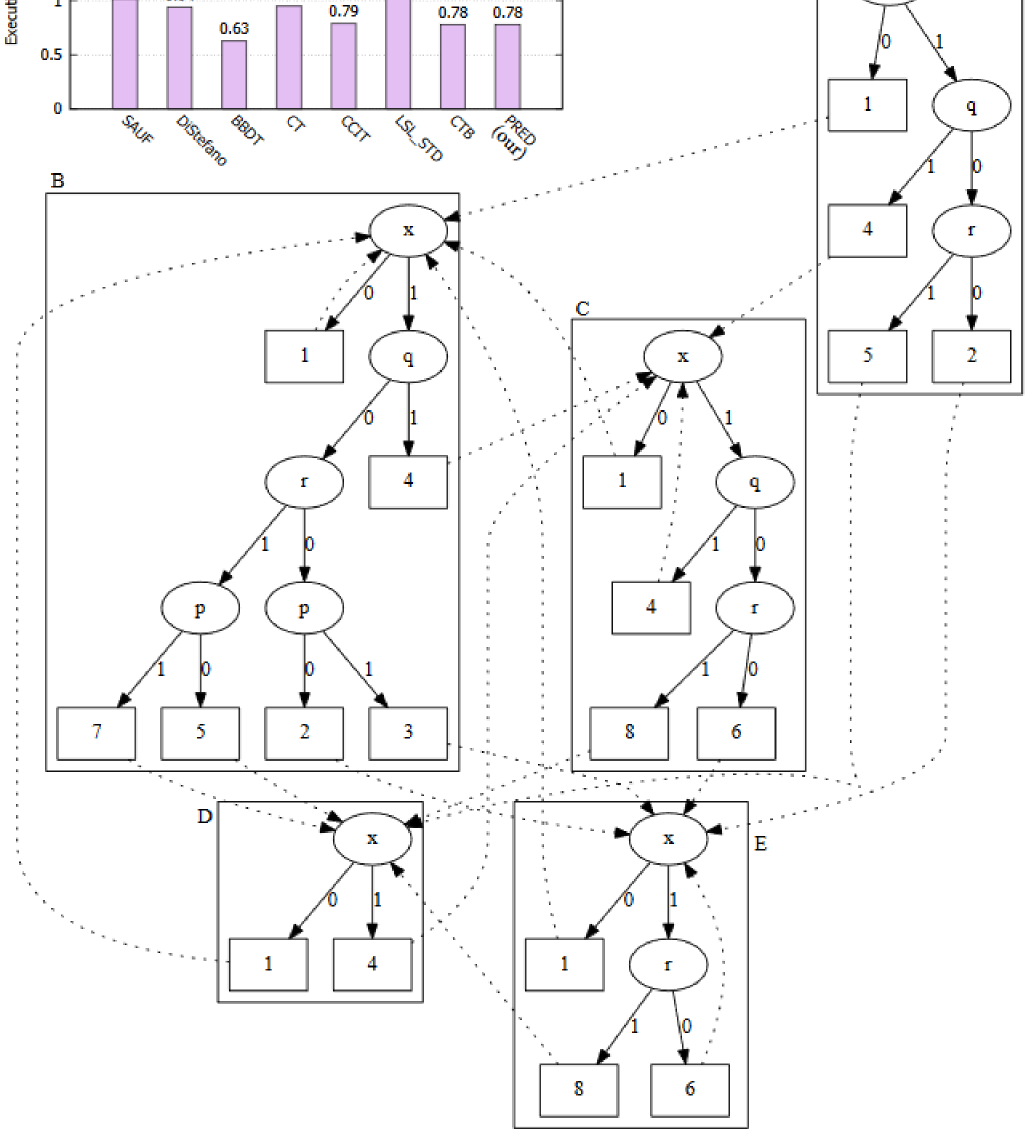






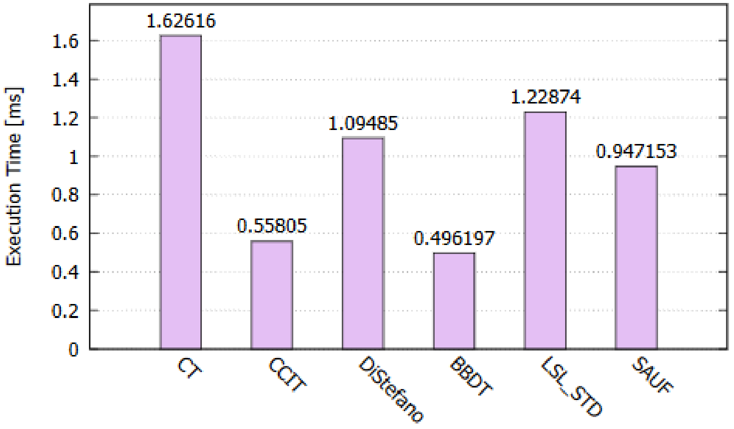

```
...
tree_C:
        if (++c >= w - 1)
                goto break_C;
        if (condition_x) {
                if (condition_q) {
                        // action 4 - x <= q
                        goto tree_C;
                }
                else {
                        if (condition_r) {
                                // action 8 - x <= r + s
                                goto tree_D;
                        }
                        else {
                                // action 6 - x <= s
                                goto tree_E;
                        }
                }
        }
        else {
                // action 1 - do nothing
                goto tree_B;
        }
...
```

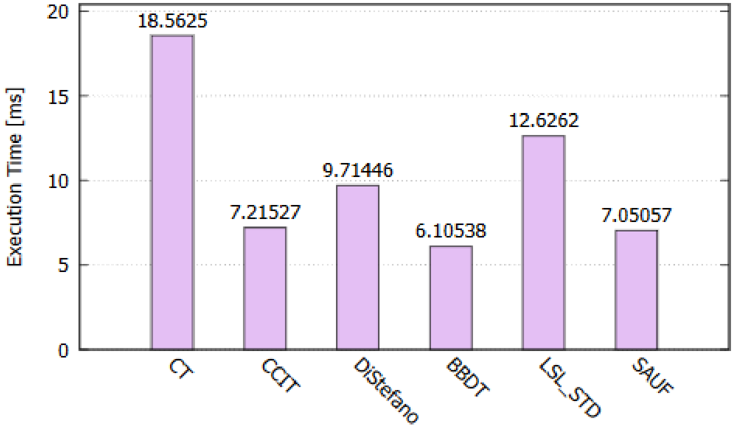


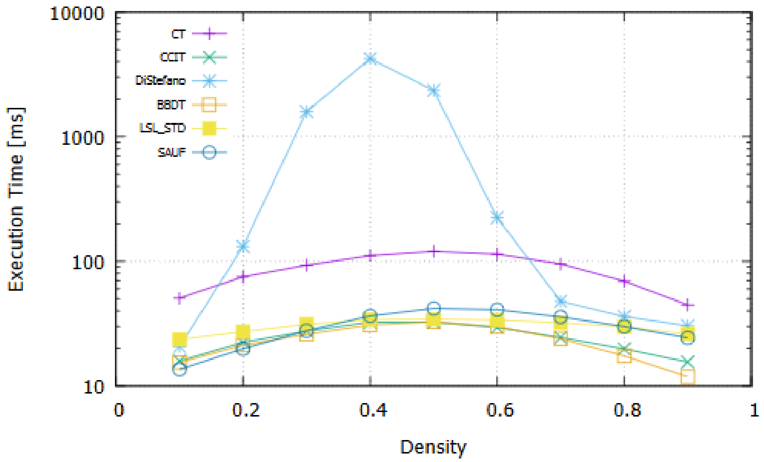


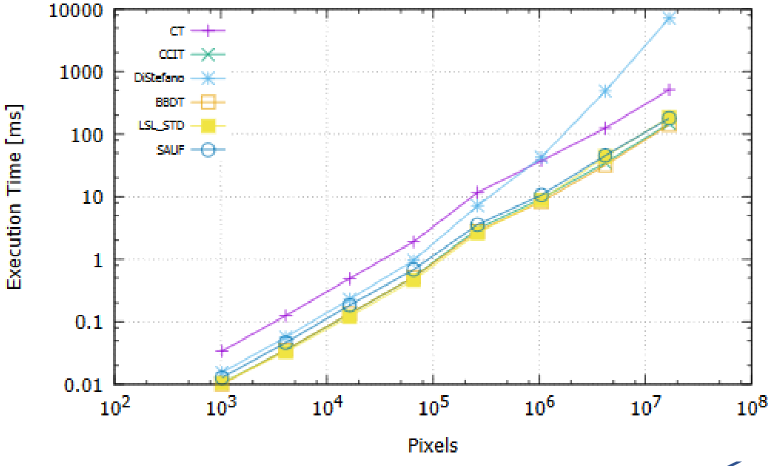
Parameter name	Description
perform	Dictionary which specifies the kind of tests to perform (correctness, average, average_ws, density and size, granularity and memory).
correctness_tests	Dictionary indicating the kind of correctness tests to perform.
tests_number	Dictionary which sets the number of runs for each test available.
Algorithms	
algorithms	List of algorithms on which apply the chosen tests.
Datasets	
check_datasets	List of datasets on which CCL algorithms should be checked.
average_datasets	List of datasets on which average test should be run.
average_ws_datasets	List of datasets on which aver-age_ws test should be run.
memory_datasets	List of datasets on which memory test should be run.
Utilities	
paths	Dictionary with both input (datasets) and output (results) paths.
write_n_labels	Whether to report the number of connected components in the output files.
color_labels	Whether to output a colored version of labeled images during tests.
save_middle_tests	Dictionary specifying, separately for every test, whether to save the output of single runs, or only a summary of the whole test.

	CT	CCIT	DiStefano	BBDT	LSL	SAUF
MIRflickr	1.62	0.55	1.09	0.49	1.22	0.94
Tob800	27.52	11.78	14.49	9.56	30.76	13.39
3DPeS	1.97	0.80	1.06	0.72	1.44	0.86
Hamlet	18.56	7.21	9.71	6.10	12.62	7.05









Parameter name	Description
perform	Dictionary which specifies the kind of tests to perform (correctness, average, average_ws, density and size, granularity and memory).
correctness_tests	Dictionary indicating the kind of correctness tests to perform.
tests_number	Dictionary which sets the number of runs for each test available.
Algorithms	
algorithms	List of algorithms on which apply the chosen tests.
Datasets	
check_datasets	List of datasets on which CCL algorithms should be checked.
average_datasets	List of datasets on which average test should be run.
average_ws_datasets	List of datasets on which average_ws test should be run.
memory_datasets	List of datasets on which memory test should be run.
Utilities	
paths	Dictionary with both input (datasets) and output (results) paths.
write_n_labels	Whether to report the number of connected components in the output files.
color_labels	Whether to output a colored version of labeled images during tests.
save_middle_tests	Dictionary specifying, separately for every test, whether to save the output of single runs, or only a summary of the whole test.

	CT	CCIT	DiStefano	BBDT	LSL	SAUF
MIRflickr	1.62	0.55	1.09	0.49	1.22	0.94
Tob800	27.52	11.78	14.49	9.56	30.76	13.39
3DPeS	1.97	0.80	1.06	0.72	1.44	0.86
Hamlet	18.56	7.21	9.71	6.10	12.62	7.05