Learning A Meta-Ensemble Technique For Skin Lesion Classification And Novel Class Detection

ISIC Skin Image Analysis Workshop, June 15th, 2020

Subhranil Bagchi Anurag Banerjee Deepti R. Bathula

Department of Computer Science and Engineering

Indian Institute of Technology Ropar

Problem Statement

- **The ISIC Challenge**[1]

- Predicting Images of Categories: *Melanoma*, *Melanocytic nevus*, *Basal cell carcinoma*, *Actinic keratosis*, *Benign keratosis*, *Dermatofibroma*, *Vascular lesion*, *Squamous cell carcinoma*, *None of the others*

- Motivation

- Our approach: *Two-level hierarchical model*

1. Homepage of ISIC 2019 Challenge: https://challenge2019.isic-archive.com/

Challenges with the ISIC 2019 Dataset

- Multi-source acquisition
- High-dimensional, low sample-space (25,331 images)
- *Eight* training classes with disproportionate samples: MEL (4,522), NV (12,875), BCC (3,323), AK (867), BKL (2,624), DF (239), VASC (253), SCC (628)
- Test time *Novelty* detection

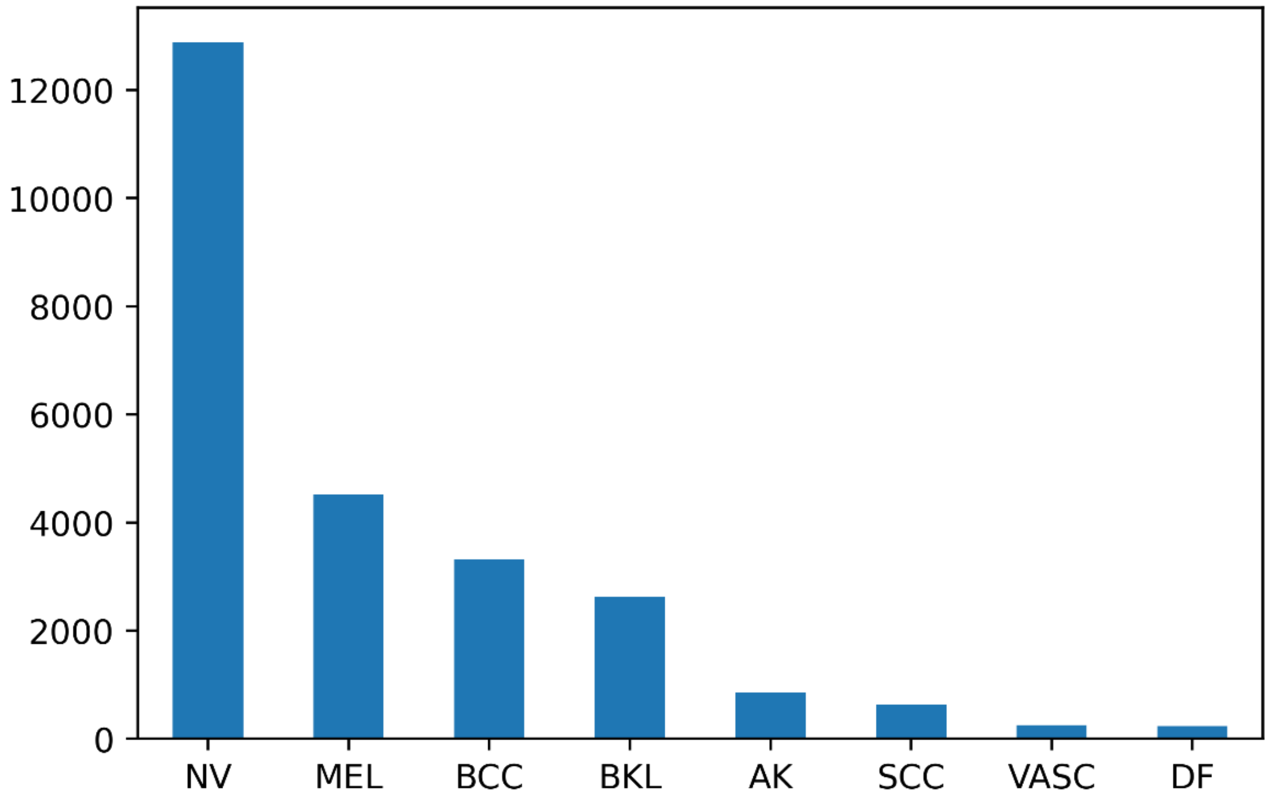


Figure: Per-class histogram depicting class imbalance for ISIC 2019 Dataset[1,2,3]

1. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions", Tschandl et. al. (2018)
2. "Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC)", Codella et. al. (2017)
3. "BCN20000: Dermoscopic Lesions in the Wild", Combalia et. al. (2019)

Preprocessing

Figure: Raw Images

Source ISIC 2019 Dataset

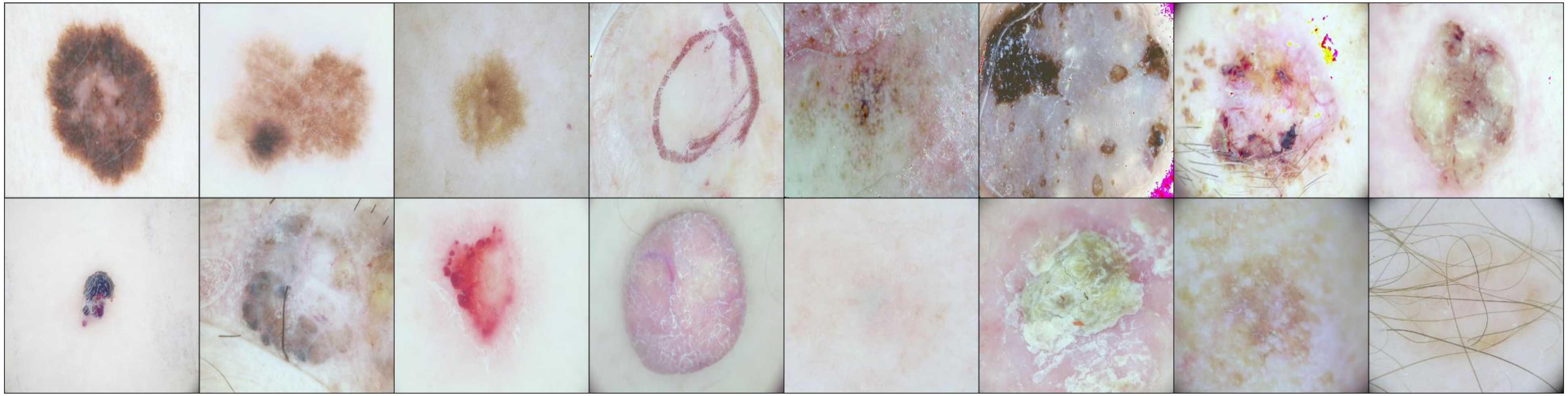


Figure: Images after preprocessing using *Shades of Gray*[1]

1. "Shades of Gray and Color Constancy", Finlayson et. al. (2004)

Stacking Module

- Pre-trained Base learners:
  - EfficientNet-B2[1]
  - EfficientNet-B5[1] (*two configurations*)
  - DenseNet-161[2]
- Meta-learner (stack of base-learners)
- Data Augmentation
- Trained with Weighted Cross-Entropy loss
- Ensemble of cross-validated models.

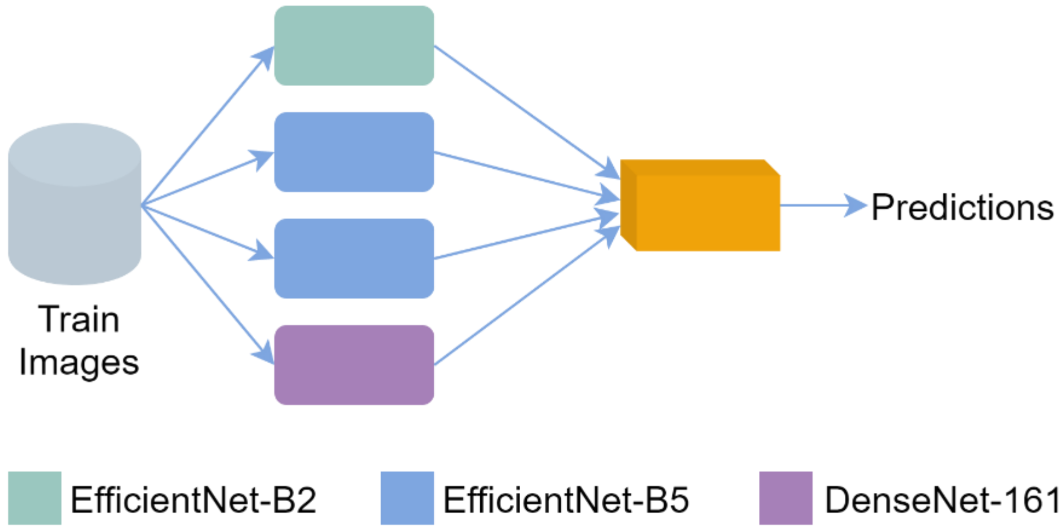


Figure: Stacking Module

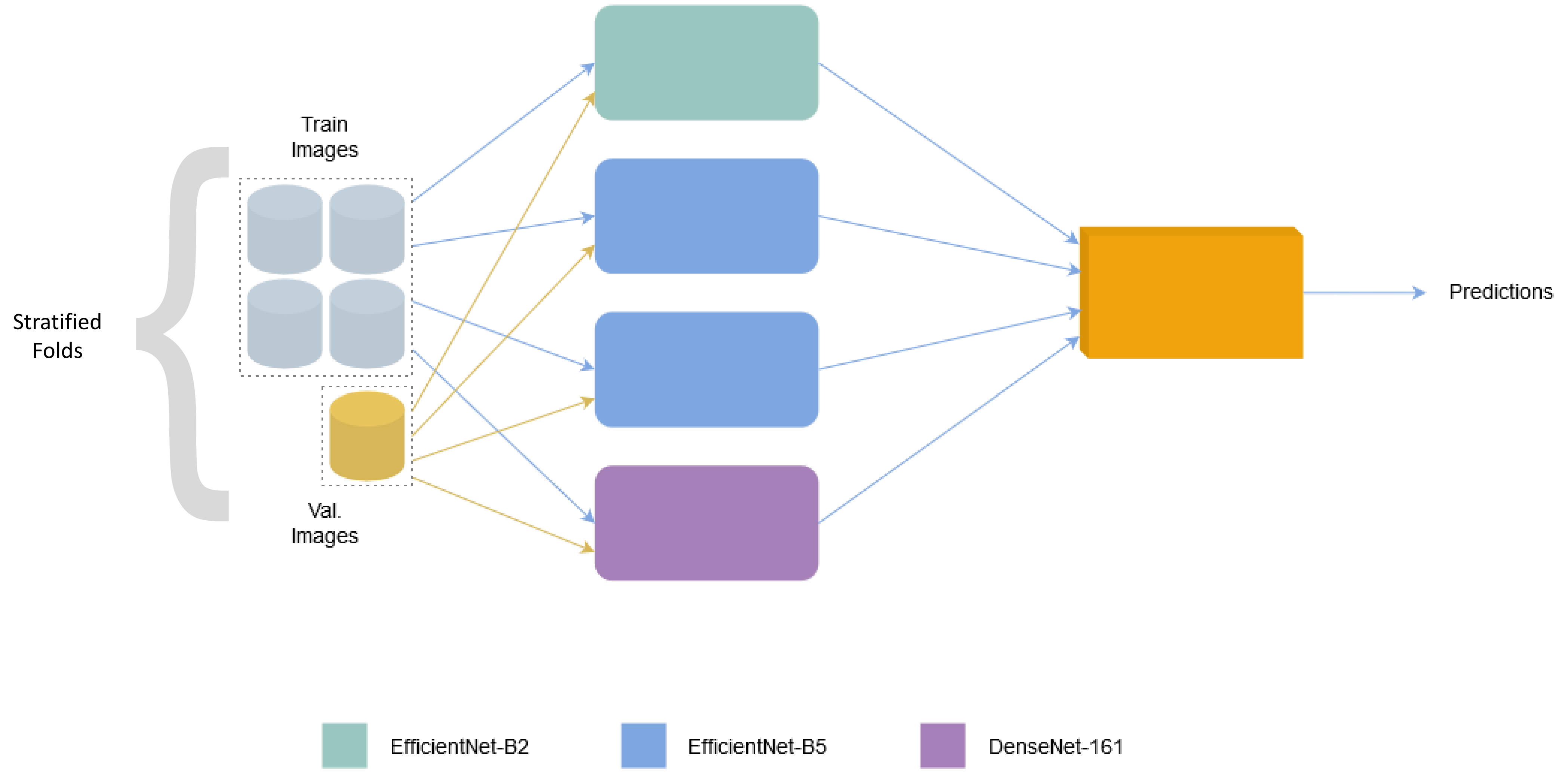
1. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, Tan et. al. (2019)
2. “Densely Connected Convolutional Networks”, Huang et. al. (2017)

Model Configuration

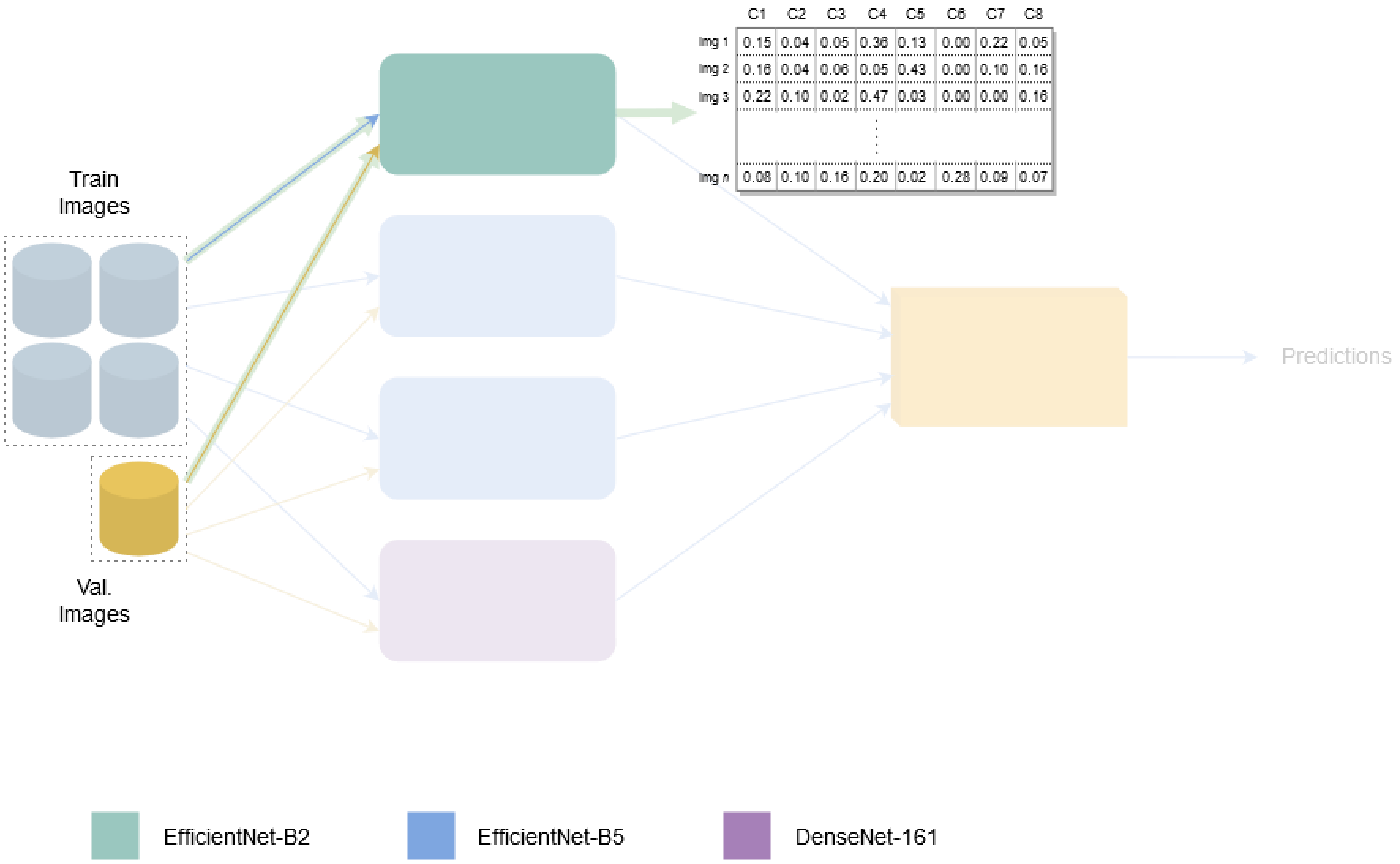
Base Model	Last Layer	Image Dim.	Crop Ratio
EfficientNet-B2	ReLU + log-SoftMax	320×320	$\frac{3}{4} \times \frac{3}{4}$
EfficientNet-B5	log-SoftMax	456×456	$\frac{3}{5} \times \frac{3}{5}$
EfficientNet-B5	ReLU + log-SoftMax	300×300	$\frac{3}{5} \times \frac{3}{5}$
DenseNet-161	log-Softmax	224×224	$\frac{3}{5} \times \frac{3}{5}$

Table: Base Learners' input configurations for Images

Stacking Module - *Training Process*

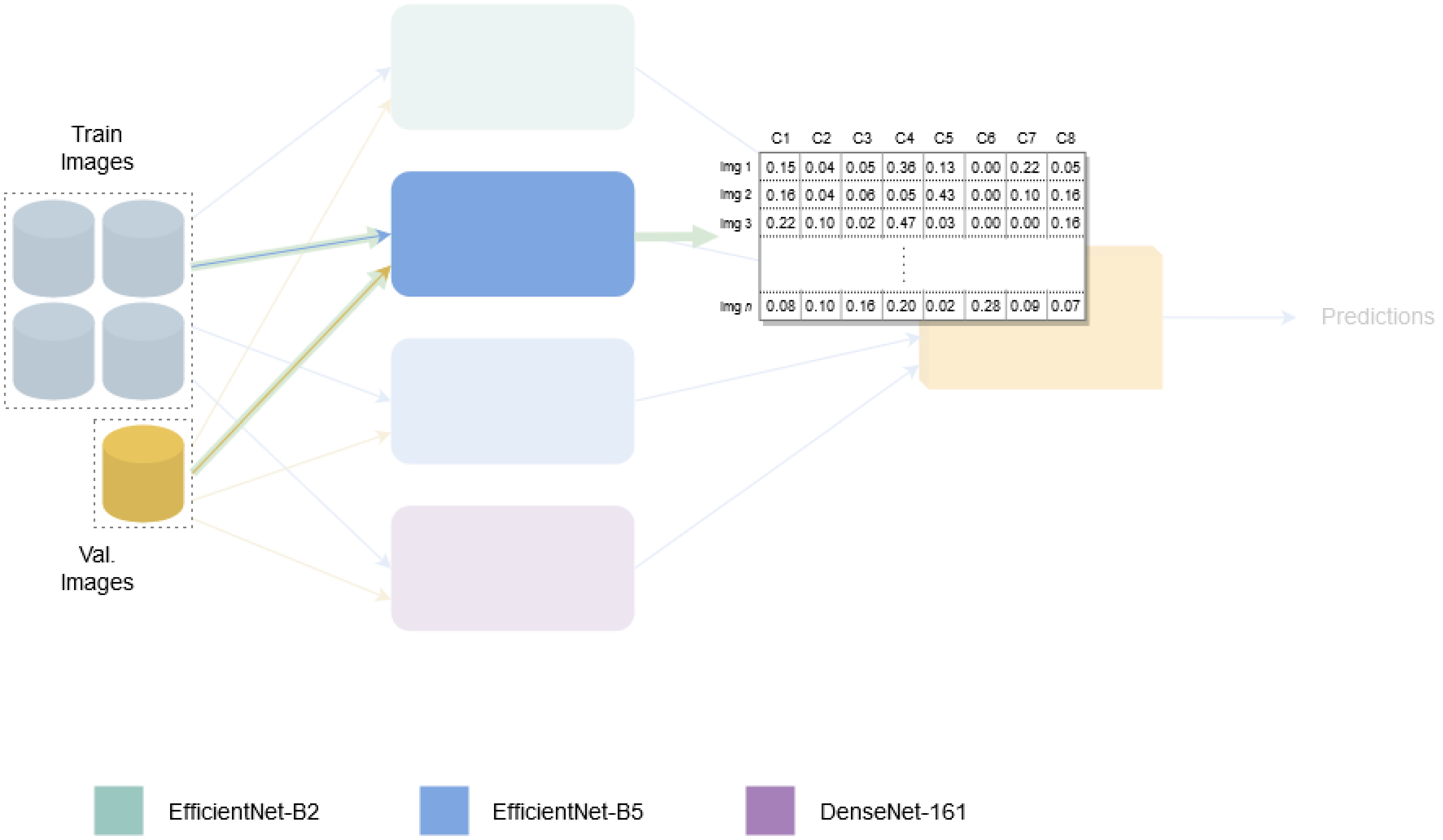


Stacking Module - *Training Process*



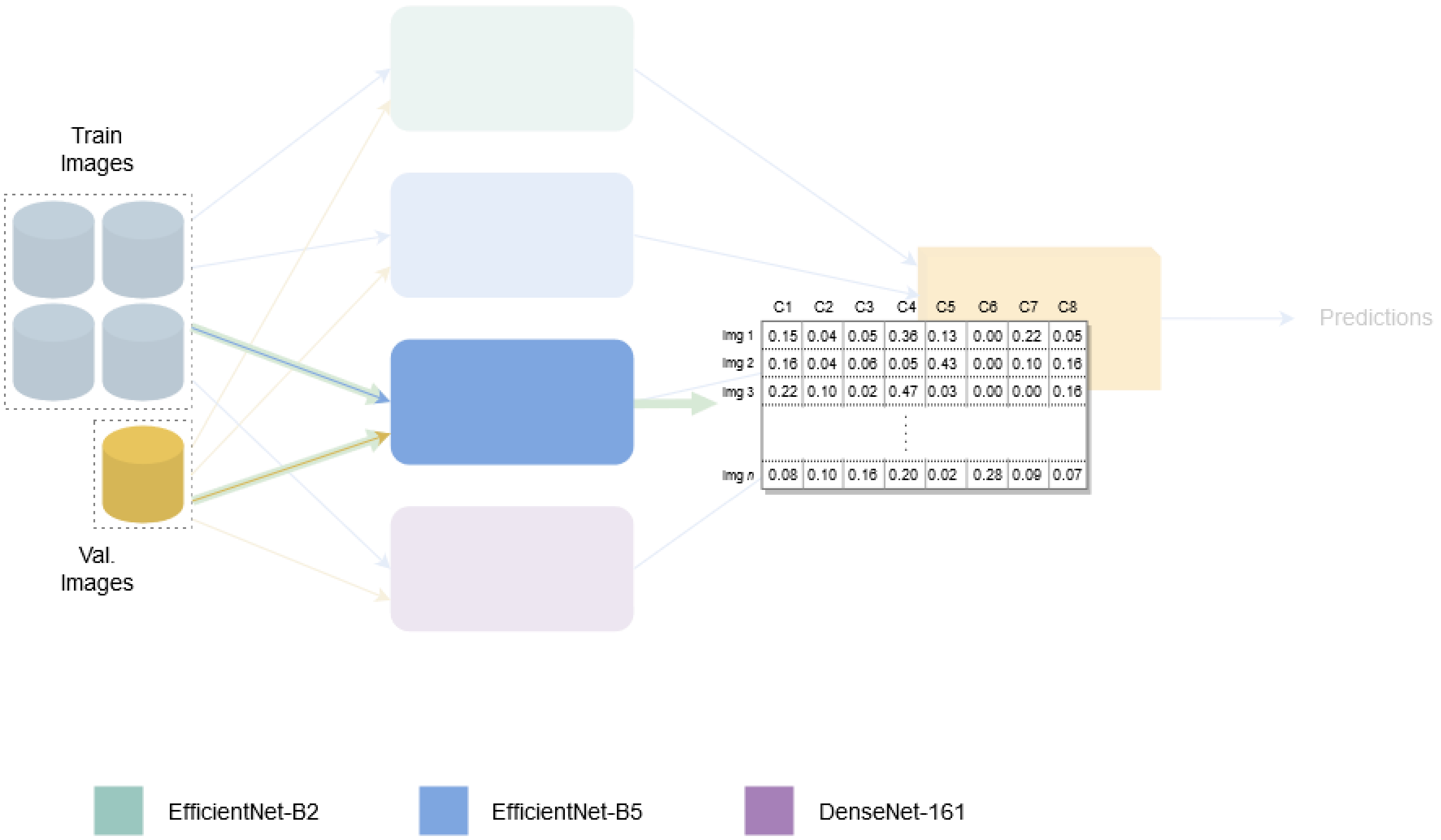
	C1	C2	C3	C4	C5	C6	C7	C8
Img 1	0.15	0.04	0.05	0.36	0.13	0.00	0.22	0.05
Img 2	0.16	0.04	0.06	0.05	0.43	0.00	0.10	0.16
Img 3	0.22	0.10	0.02	0.47	0.03	0.00	0.00	0.16
⋮								
Img *n*	0.08	0.10	0.16	0.20	0.02	0.28	0.09	0.07

Stacking Module - *Training Process*



	C1	C2	C3	C4	C5	C6	C7	C8
Img 1	0.15	0.04	0.05	0.36	0.13	0.00	0.22	0.05
Img 2	0.16	0.04	0.06	0.05	0.43	0.00	0.10	0.16
Img 3	0.22	0.10	0.02	0.47	0.03	0.00	0.00	0.16
⋮								
Img *n*	0.08	0.10	0.16	0.20	0.02	0.28	0.09	0.07

Stacking Module-*Training Process*

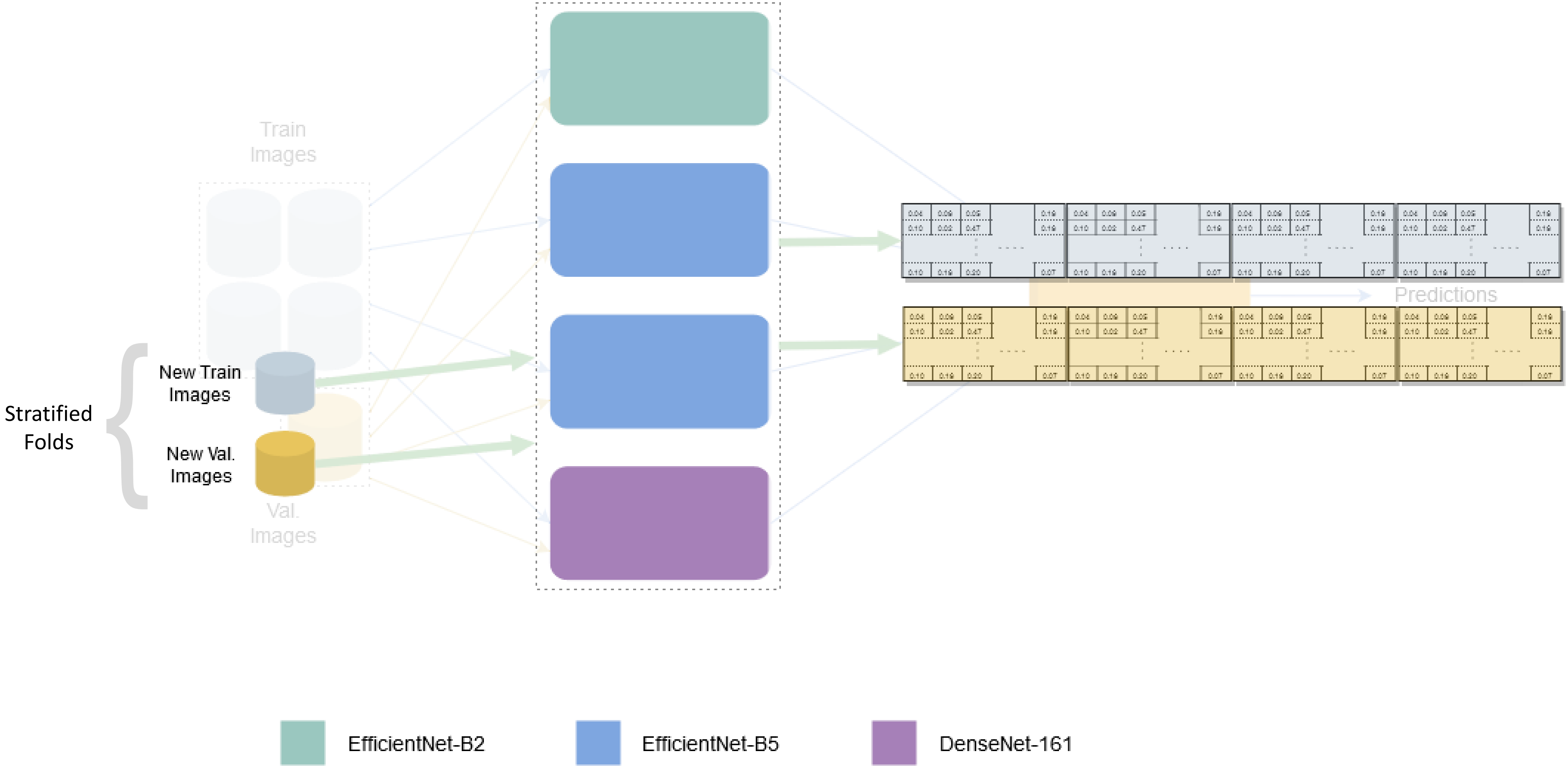


	C1	C2	C3	C4	C5	C6	C7	C8
Img 1	0.15	0.04	0.05	0.36	0.13	0.00	0.22	0.05
Img 2	0.16	0.04	0.06	0.05	0.43	0.00	0.10	0.16
Img 3	0.22	0.10	0.02	0.47	0.03	0.00	0.00	0.16
⋮								
Img *n*	0.08	0.10	0.16	0.20	0.02	0.28	0.09	0.07

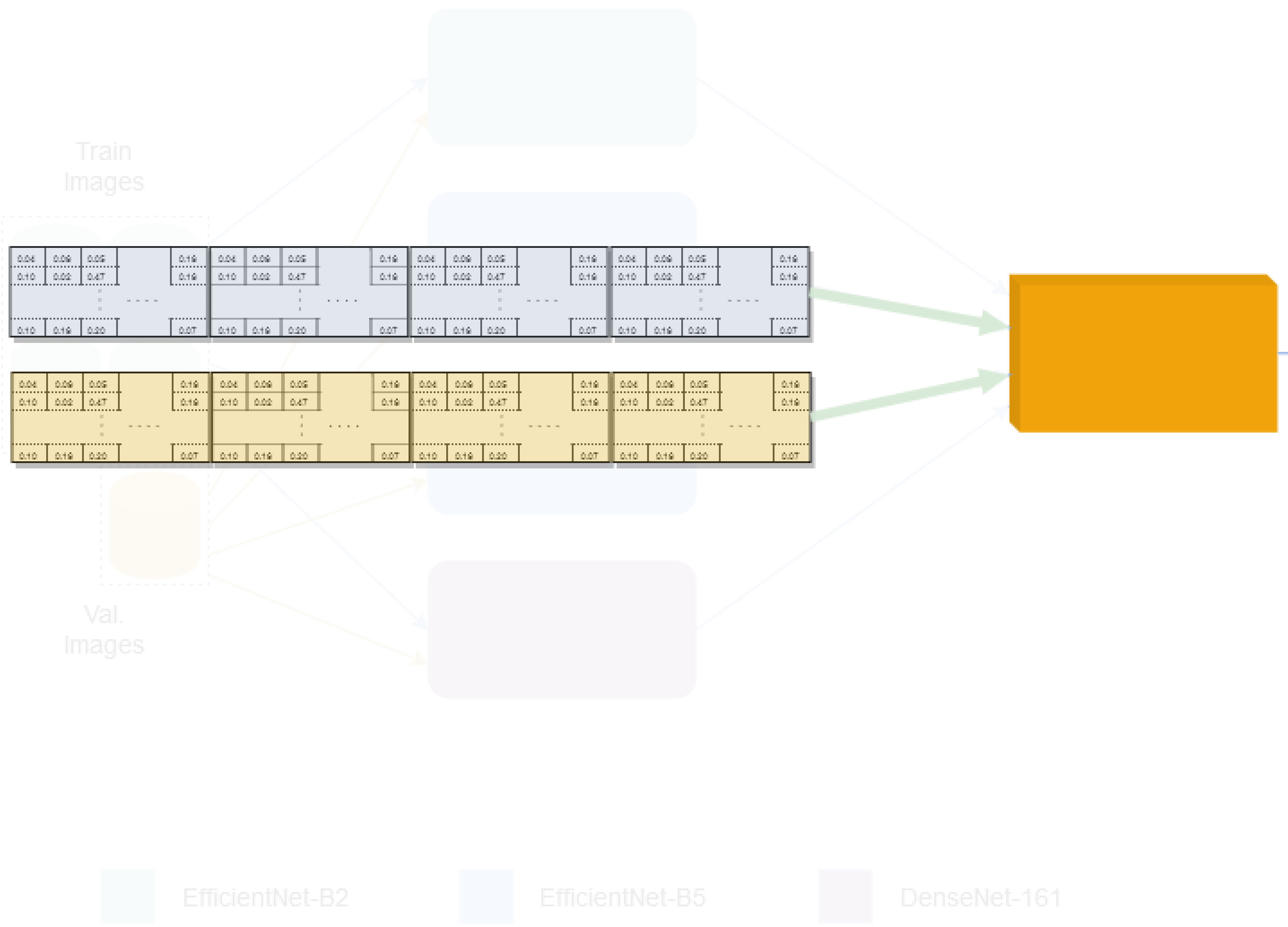
Stacking Module - *Training Process*

	C1	C2	C3	C4	C5	C6	C7	C8
Img 1	0.15	0.04	0.05	0.36	0.13	0.00	0.22	0.05
Img 2	0.16	0.04	0.06	0.05	0.43	0.00	0.10	0.16
Img 3	0.22	0.10	0.02	0.47	0.03	0.00	0.00	0.16
				⋮				
Img *n*	0.08	0.10	0.16	0.20	0.02	0.28	0.09	0.07

Stacking Module - *Training Process*



Stacking Module - *Training Process*



	C1	C2	C3	C4	C5	C6	C7	C8
Img 1	0.15	0.04	0.05	0.36	0.13	0.00	0.22	0.05
Img 2	0.16	0.04	0.06	0.05	0.43	0.00	0.10	0.16
Img 3	0.22	0.10	0.02	0.47	0.03	0.00	0.00	0.16
⋮								
Img *n*	0.08	0.10	0.16	0.20	0.02	0.28	0.09	0.07

t-SNE Plots

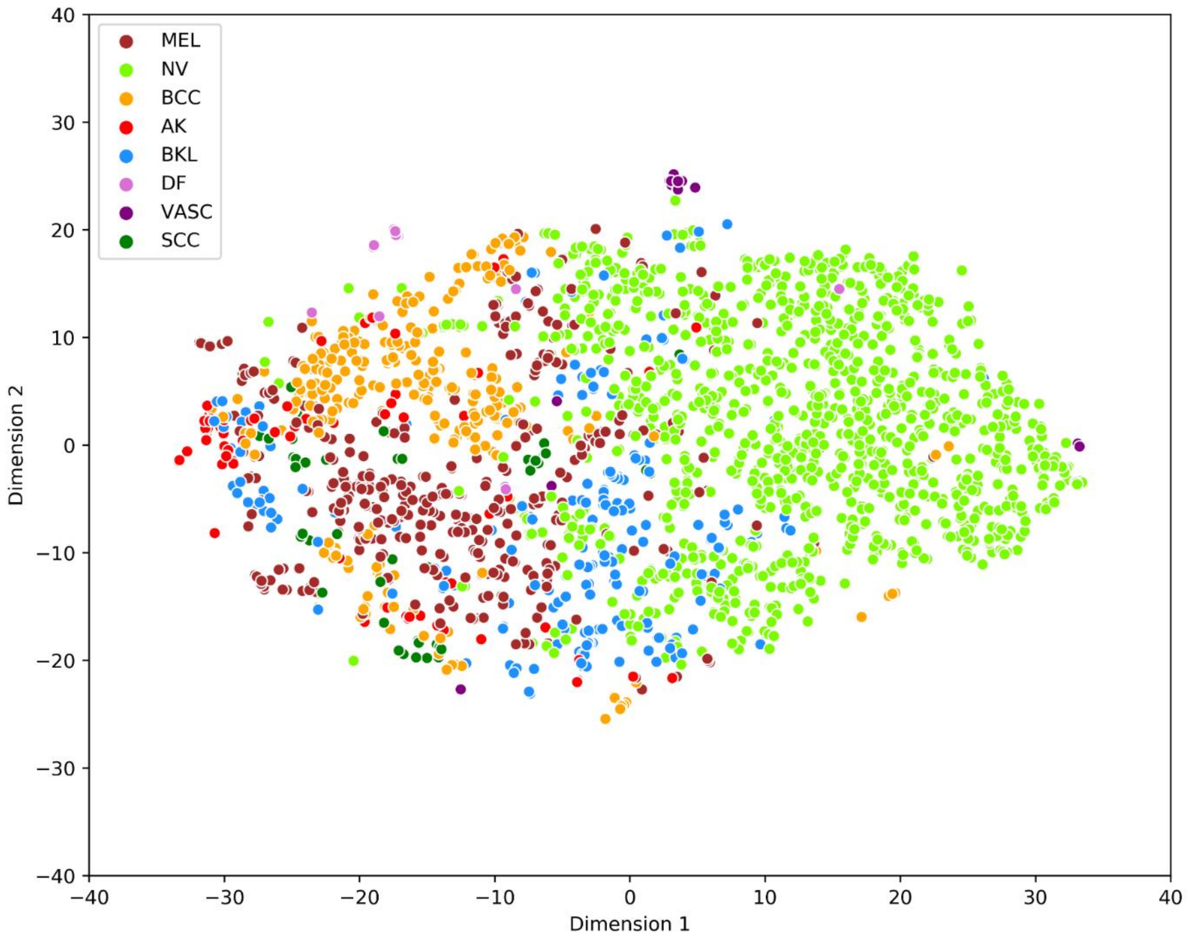


Figure: t-SNE[1,2] plot for Average Model on Validation Set- 4.2

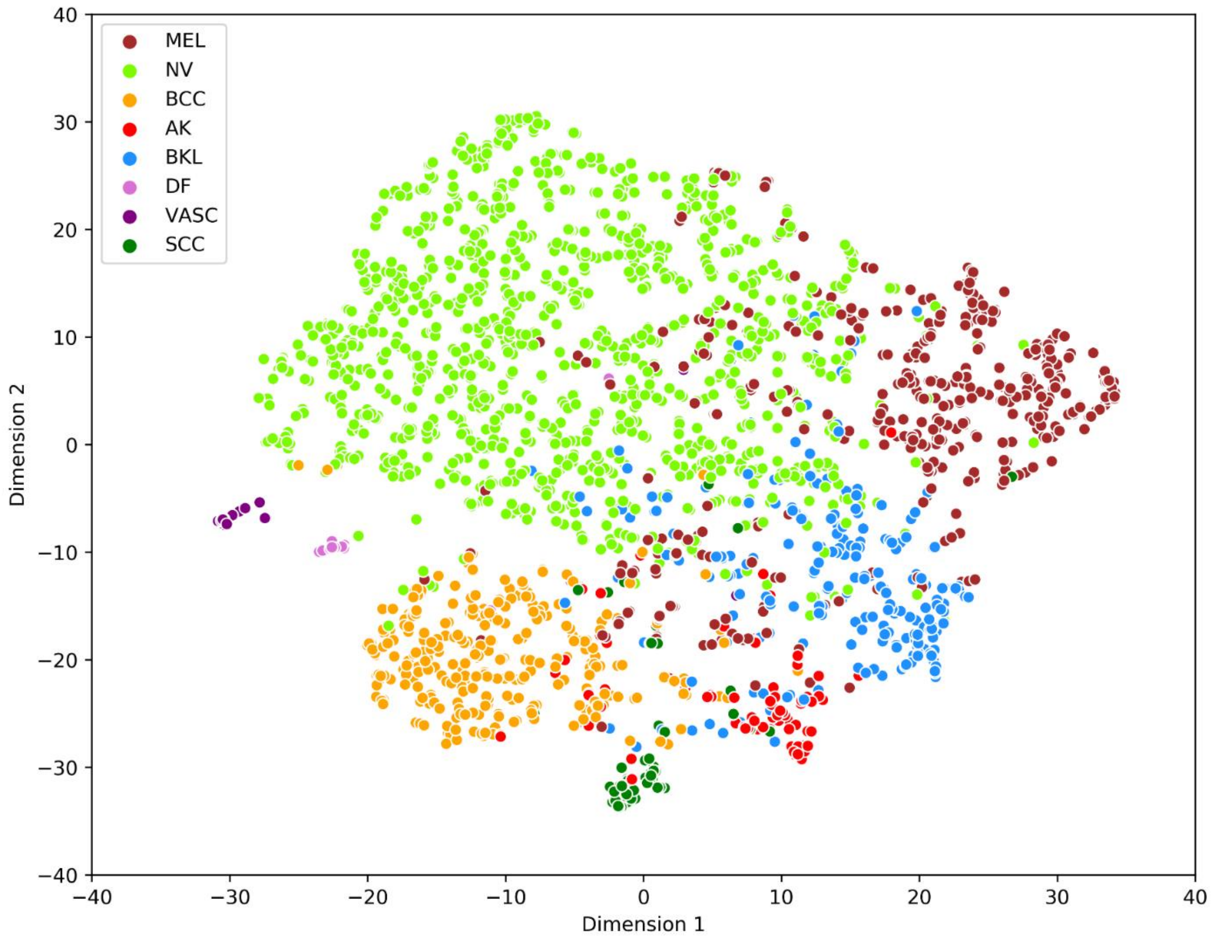


Figure: t-SNE plot for Stack Model on Validation Set- 4.2

1. "Visualizing Data using t-SNE", Maaten et. al. (2008)
2. "GPU Accelerated t-distributed Stochastic Neighbor Embedding", Chan et. Al. (2019)

t-SNE Plots (Cont.)

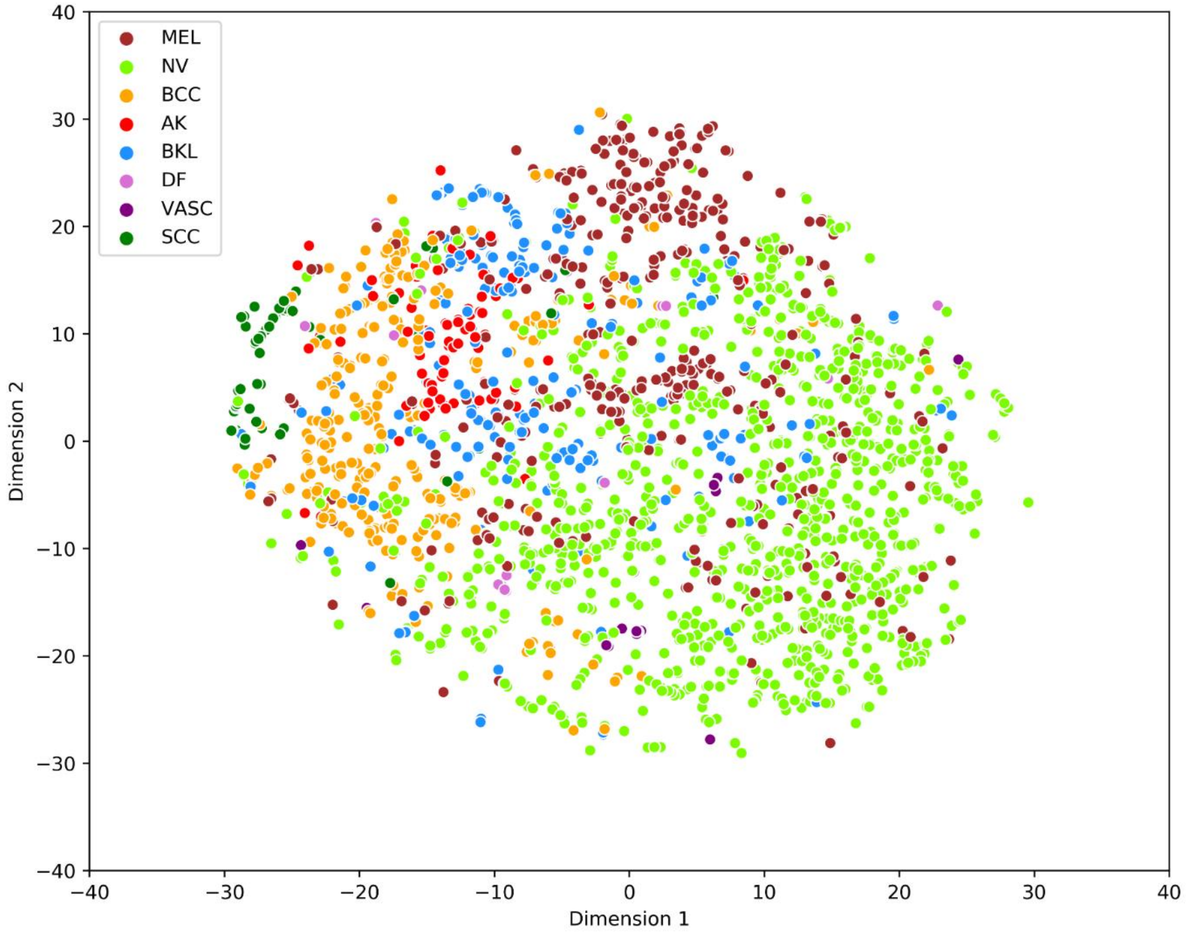


Figure: t-SNE plot for Average Model on Validation Set- 2.2

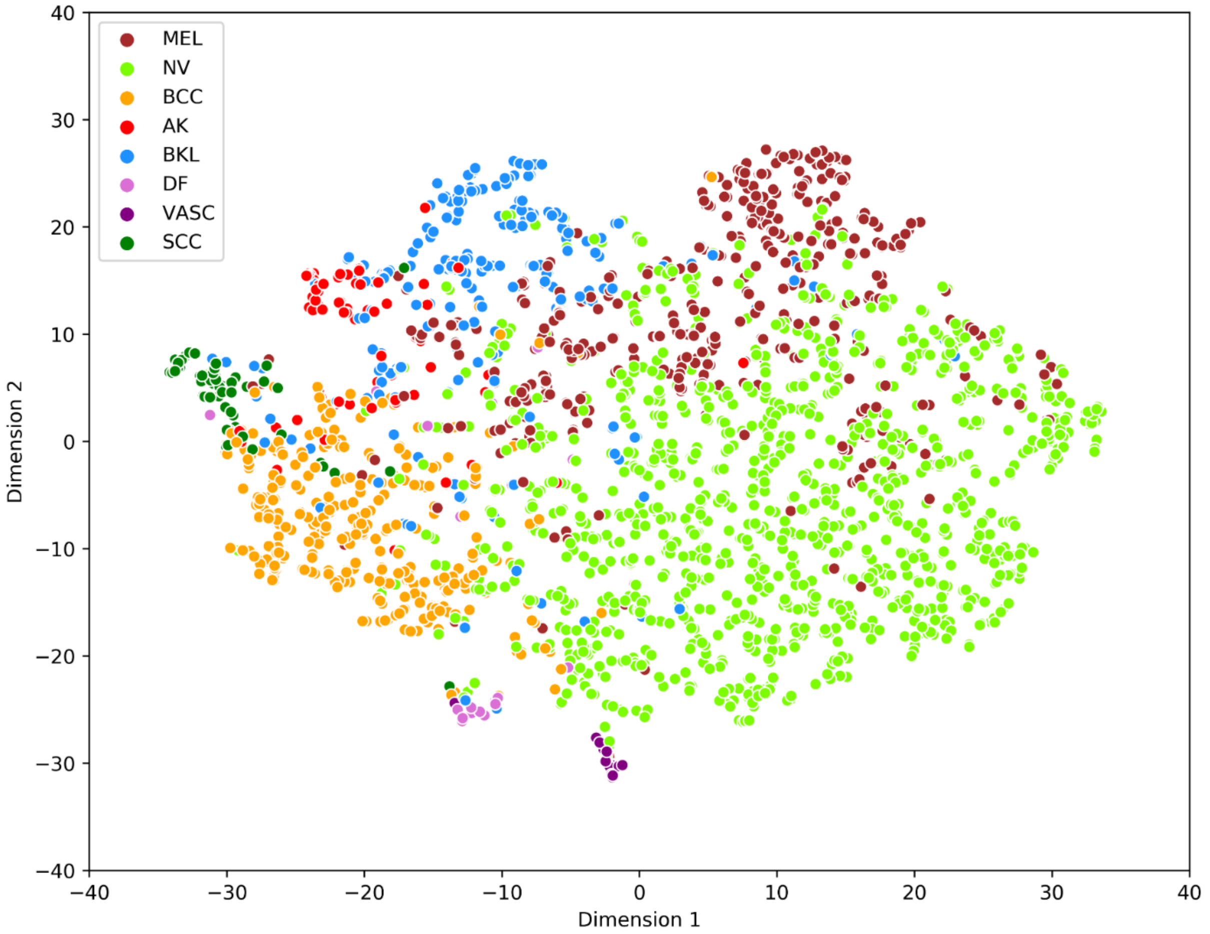


Figure: t-SNE plot for Stack Model on Validation Set- 2.2

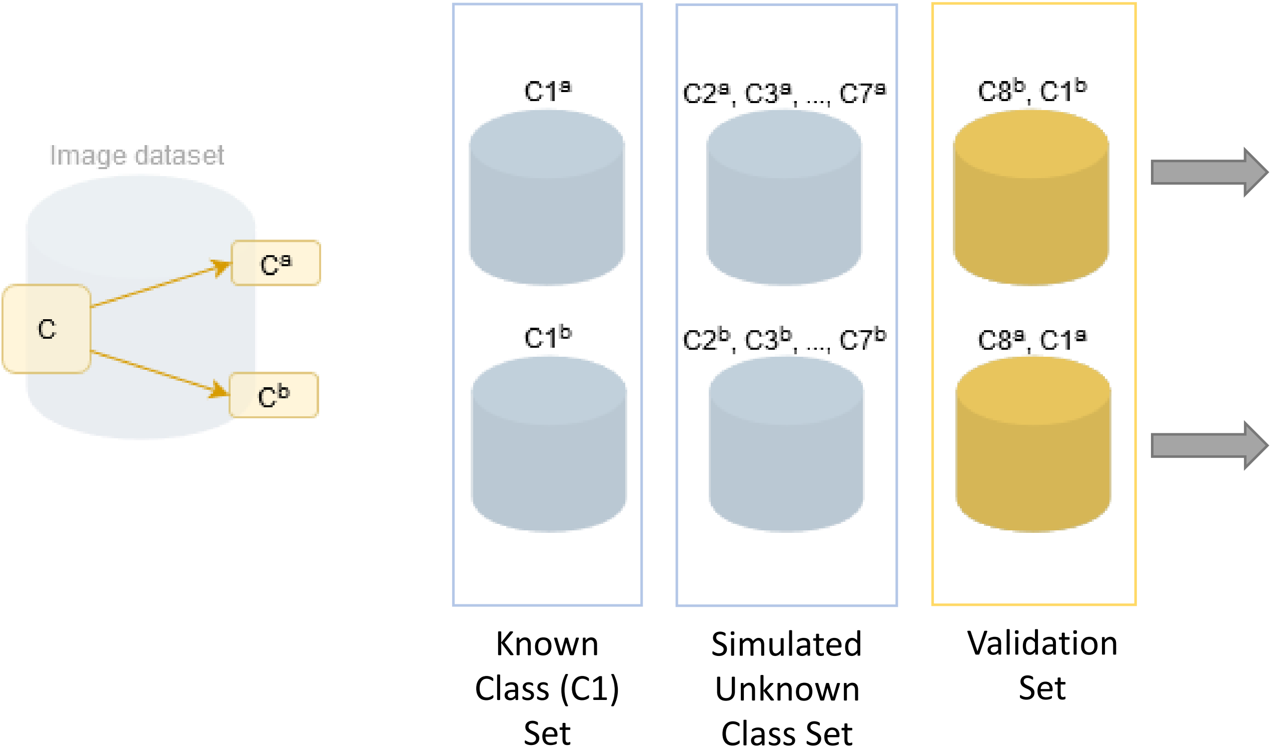
Class Specific – Known vs. Simulated Unknown Modules (CS-KSU)

- Class-wise individual modules (*one vs. rest*)
- Trained for multiple folds, (*with simulated unknowns*)
- ResNet-18[1]
- Data Augmentation
- Trained with Weighted Cross-Entropy and Triplet Loss
- Prediction average
- Thresholding

1. "Deep Residual Learning for Image Recognition", He et. al. (2016)

Class Specific – Known vs. Simulated Unknown Modules – *The Splits*

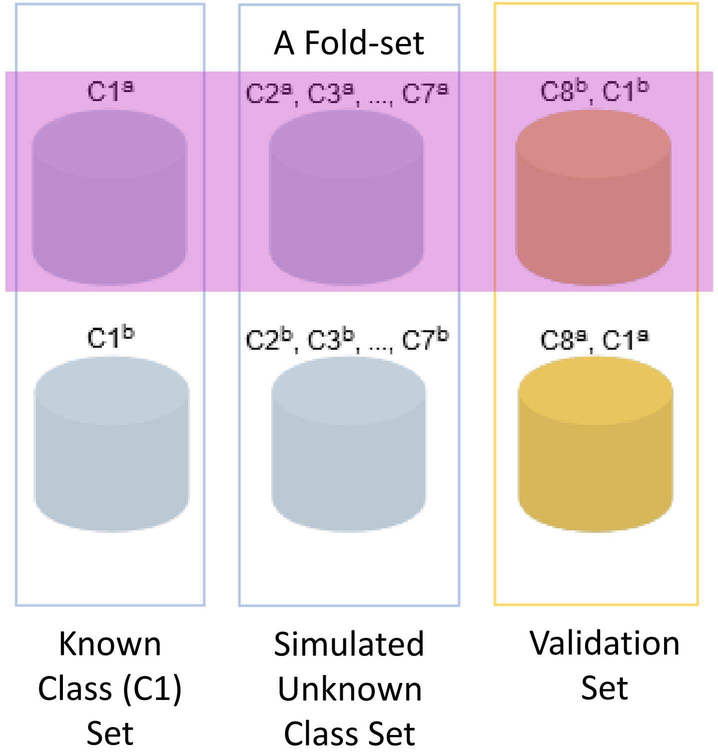
- Trained with leave-one-*unknown-class*-out, one-*versus*-rest cross validation



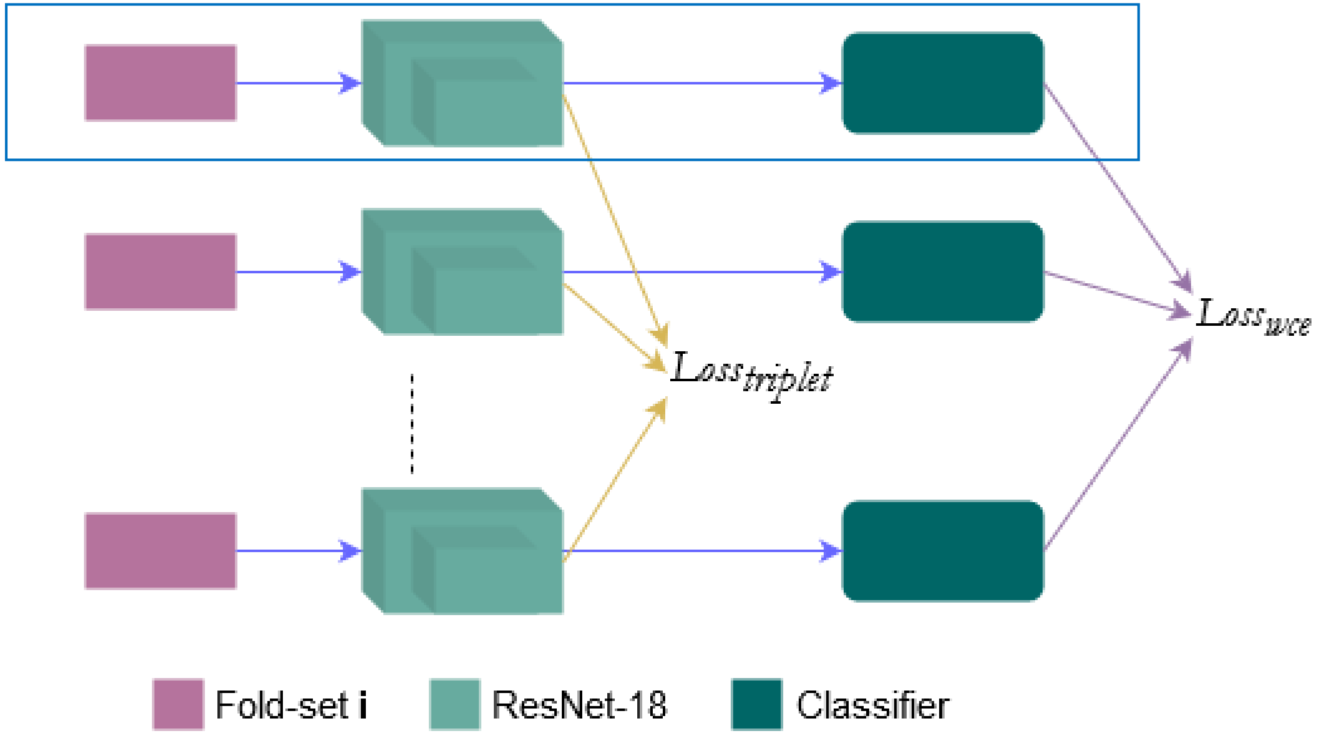
7 Combinations for the Simulated Unknown Class Set and Validation Set
e.g. {$C2^b$, $C3^b$, …, **$C8^b$**} {**$C7^b$**, $C1^b$}

7 Combinations for the Simulated Unknown Class Set and Validation Set
e.g. {$C2^a$, $C3^a$, …, **$C8^a$**} {**$C7^a$**, $C1^a$}

Class Specific – Known vs. Simulated Unknown Modules – *The Splits*

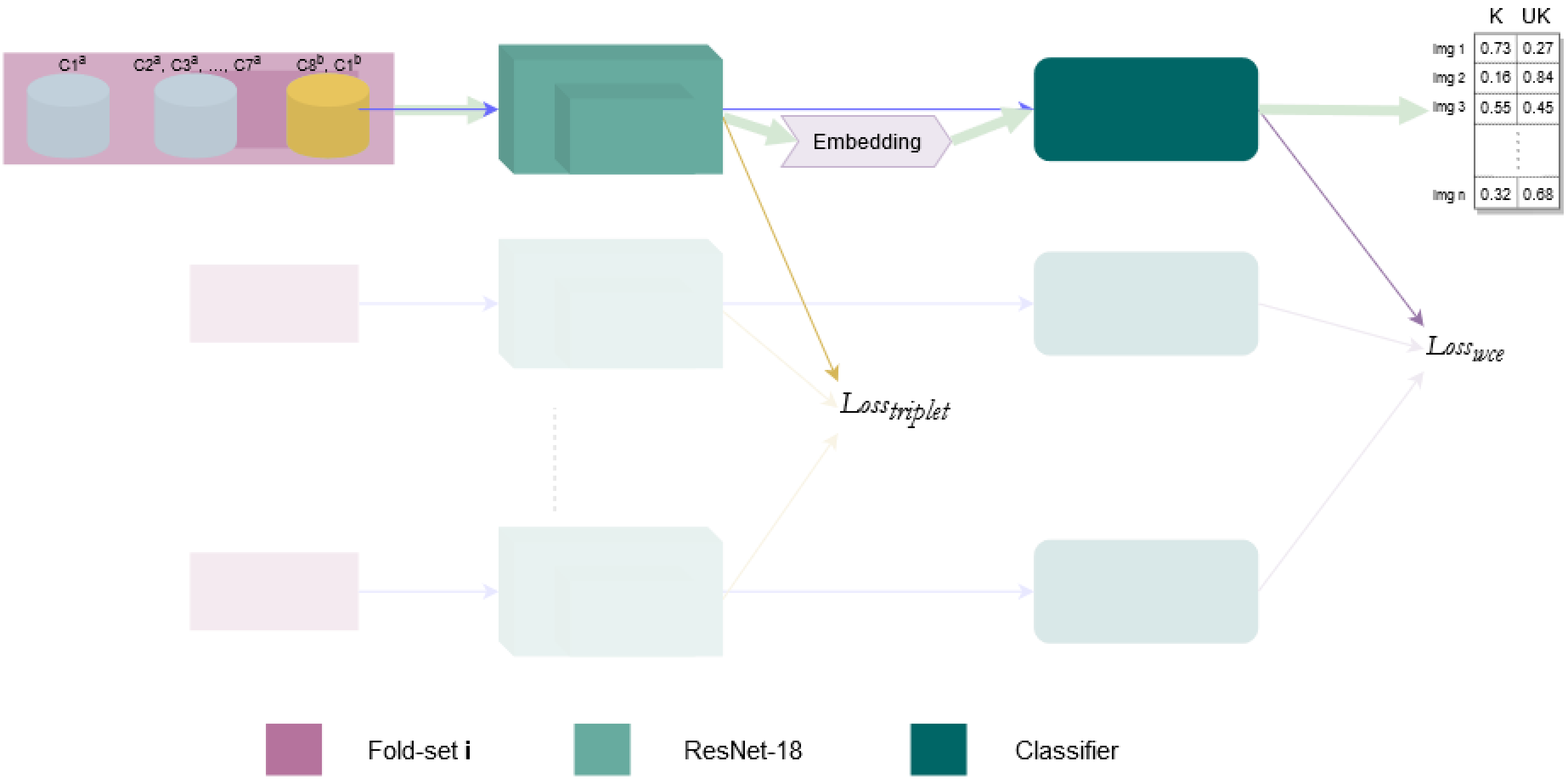


Class Specific – Known vs. Simulated Unknown Modules – *Training Process*



14 Models per Known Class (i.e., per CS-KSU Module)

Class Specific – Known vs. Simulated Unknown Modules – *Training Process*



Thresholding Explained

	Known Class	Un-known Class	Predicted Classes for Thresholds					Ground Truth
			0.025	0.050	0.075	...	0.975	
Fold-set 1	0.73	0.27	K	K	K		UK	K
	0.16	0.84	K	K	K		UK	UK
	0.55	0.45	K	K	K	...	UK	UK
	...		...					...
	0.32	0.68	K	K	K		UK	K
...								
Fold-set 14	0.96	0.04	K	UK	UK		UK	UK
	0.89	0.11	K	K	K		UK	UK
	0.67	0.33	K	K	K	...	UK	K
	...		...					...
	0.42	0.58	K	K	K		UK	K
Balanced Accuracy			0.2	0.6	$\boxed{0.8}$		0.3	

Choice for Cost Functions

Weighted Cross Entropy Loss[1]

- Deals with imbalanced class distribution

$$\mathcal{L}_{wce} = -\frac{1}{N}\sum_{c=1}^{C}\sum_{n=1}^{N} w_c \times y_n^c \times \log\left(h_\theta\left(x_n, c\right)\right)$$

where,

N = Total number of training examples
C = Total number of classes
w_c = Weight for class c
y_n^c = Target label for training example n of class c
x_n = Input for training example n
h_θ = Some model with weight parameter θ

1. "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling", Ho et. al. (2020)

Choice for Cost Functions

Triplet Loss[1]

- Reduces distance between same class samples, whereas broadens otherwise

- Useful for margin in latent space between known and simulated unknowns

$$\mathcal{L}(A, B, Y) = max\Big(dist(A, B) - dist(A, Y) + \gamma, 0\Big)$$

where,

A is the anchor point embedding
B is the embedding of an instance in same class as the anchor
Y is the embedding of an instance not in anchor's class
γ is a margin between positive and negative pairs
$dist()$ is some distance metric function

1. "FaceNet: A Unified Embedding for Face Recognition and Clustering", Scroff et. al. (2015)

Testing Process – Complete Model

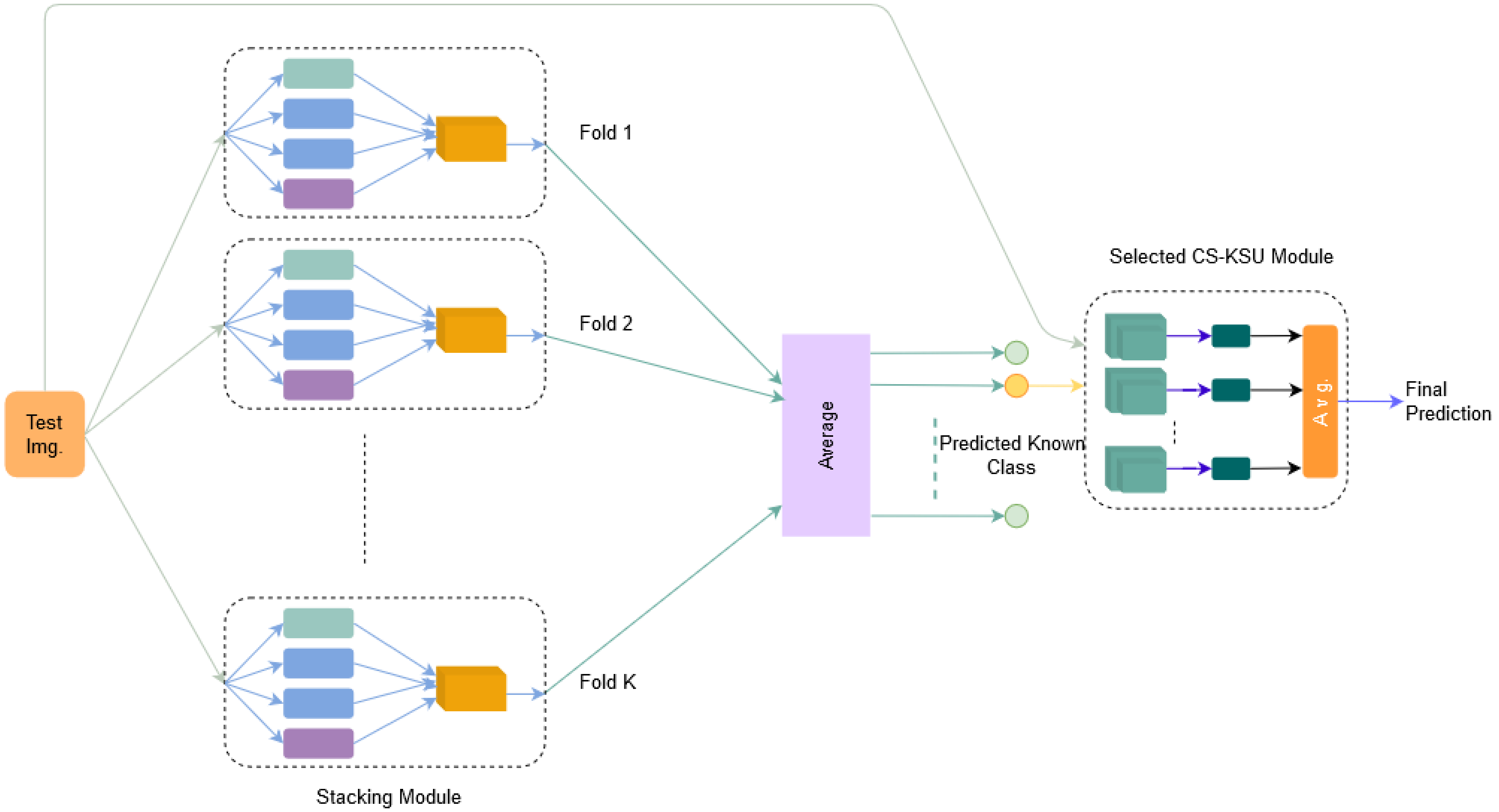
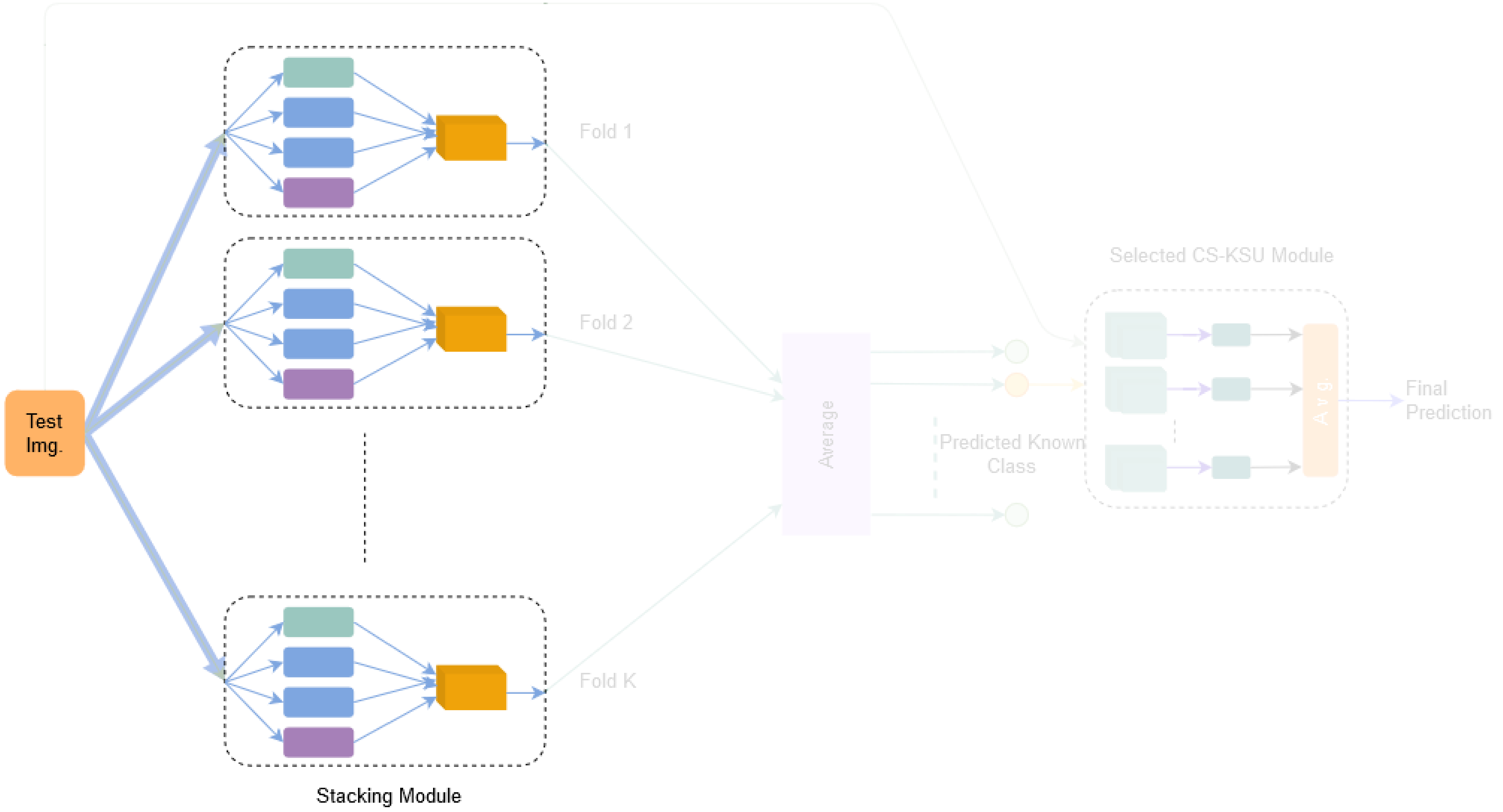
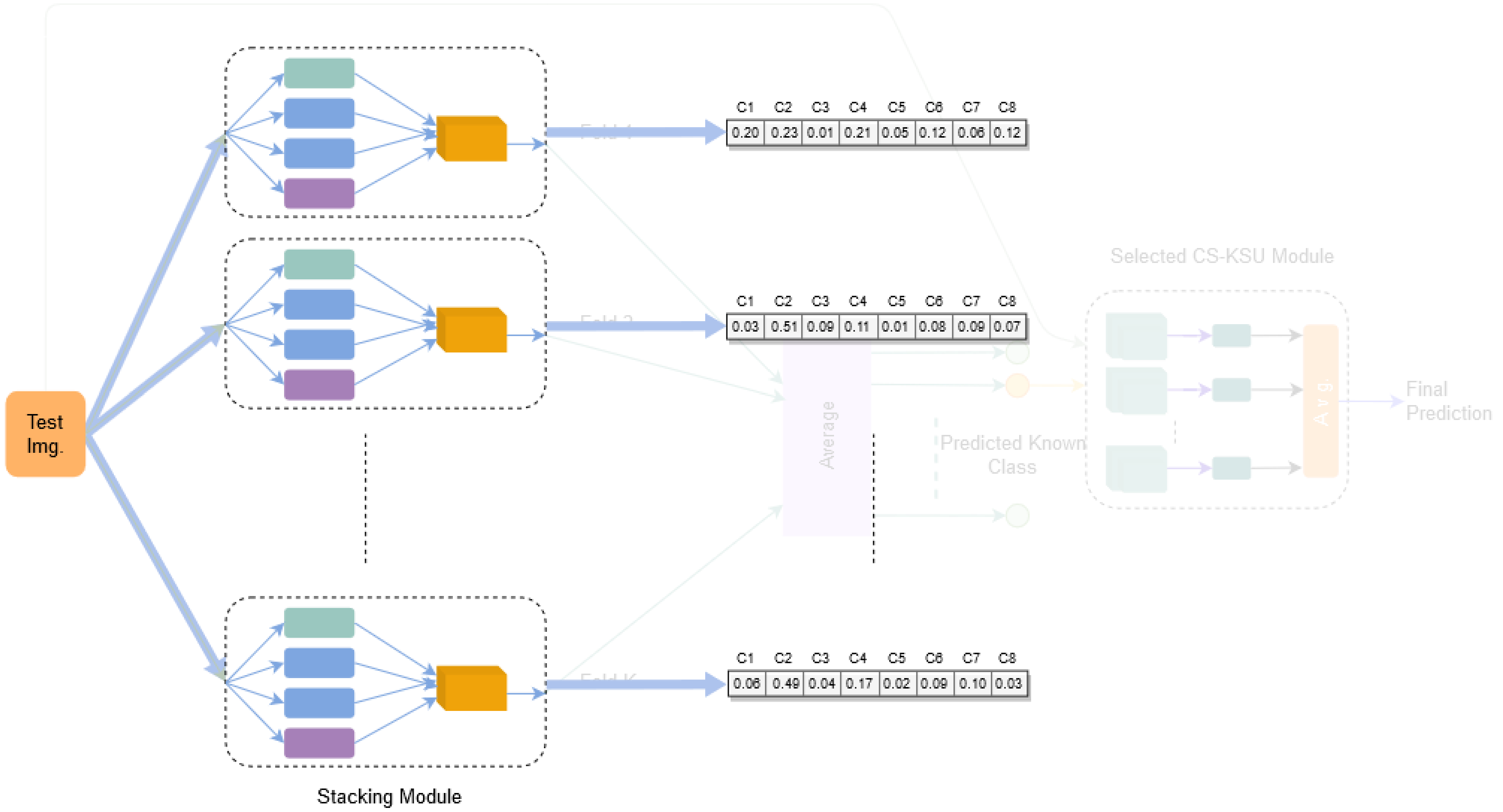


Figure: Diagram explaining the testing procedure

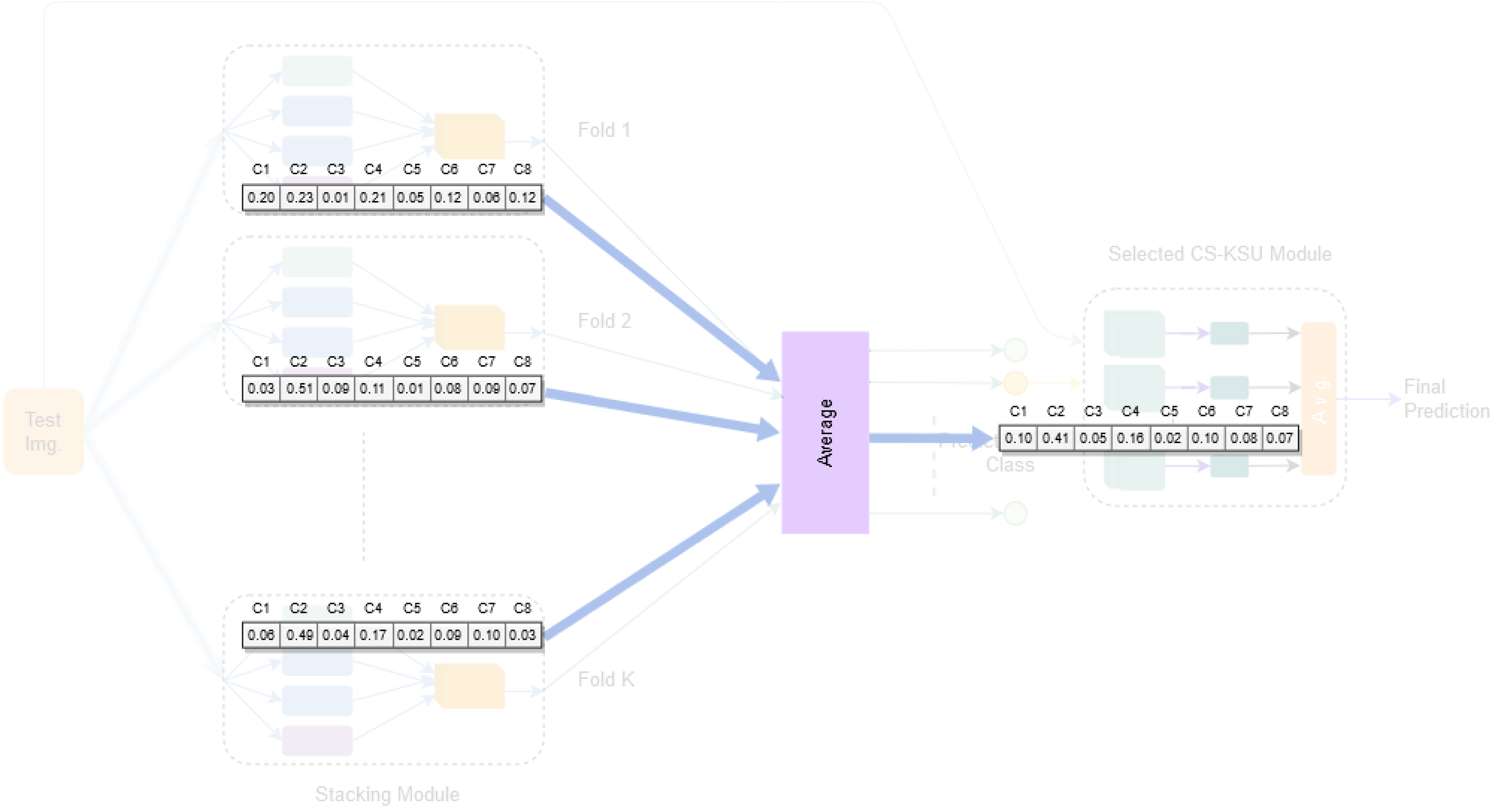
Testing Process – Explained



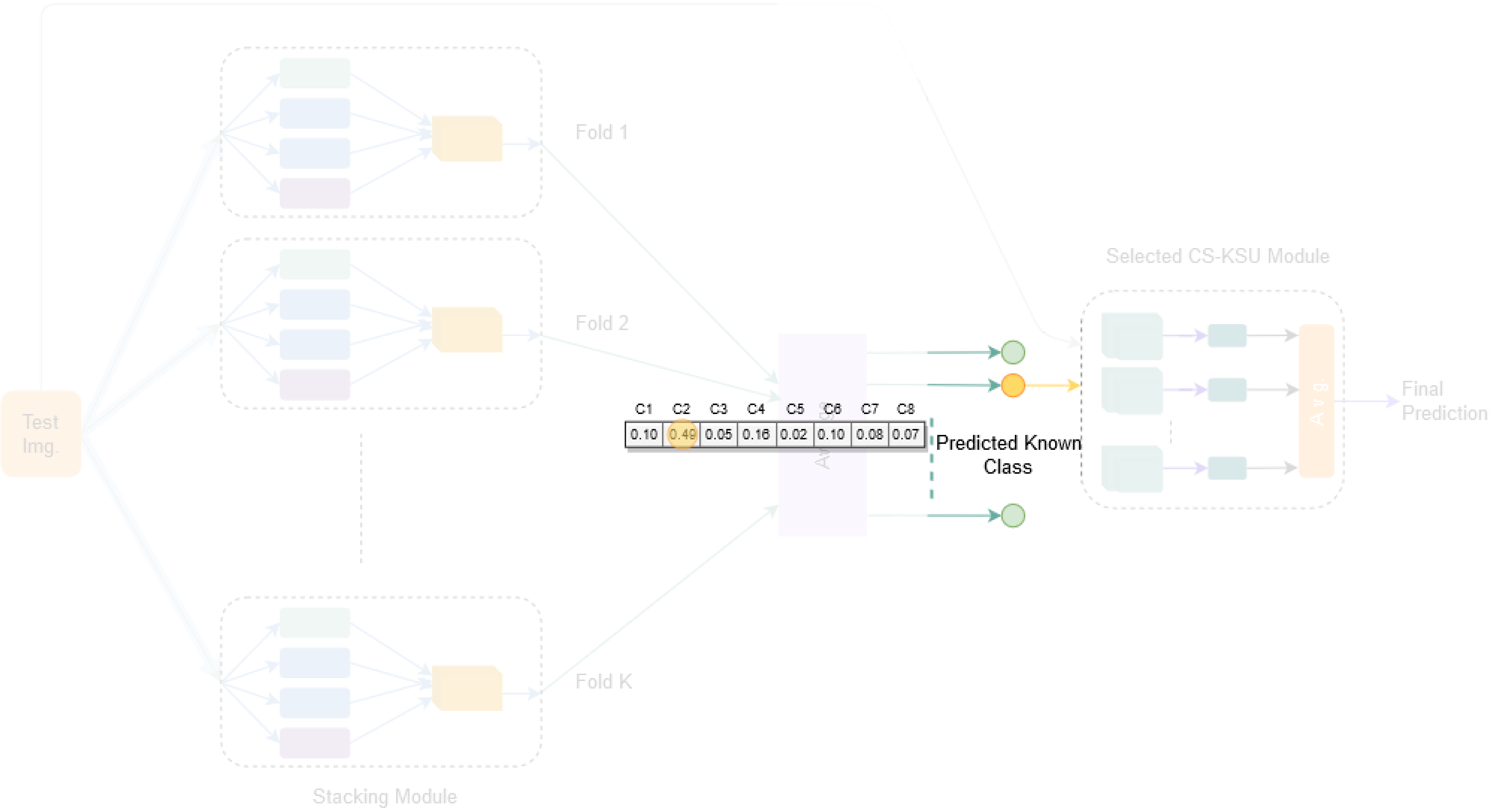
Testing Process – Explained



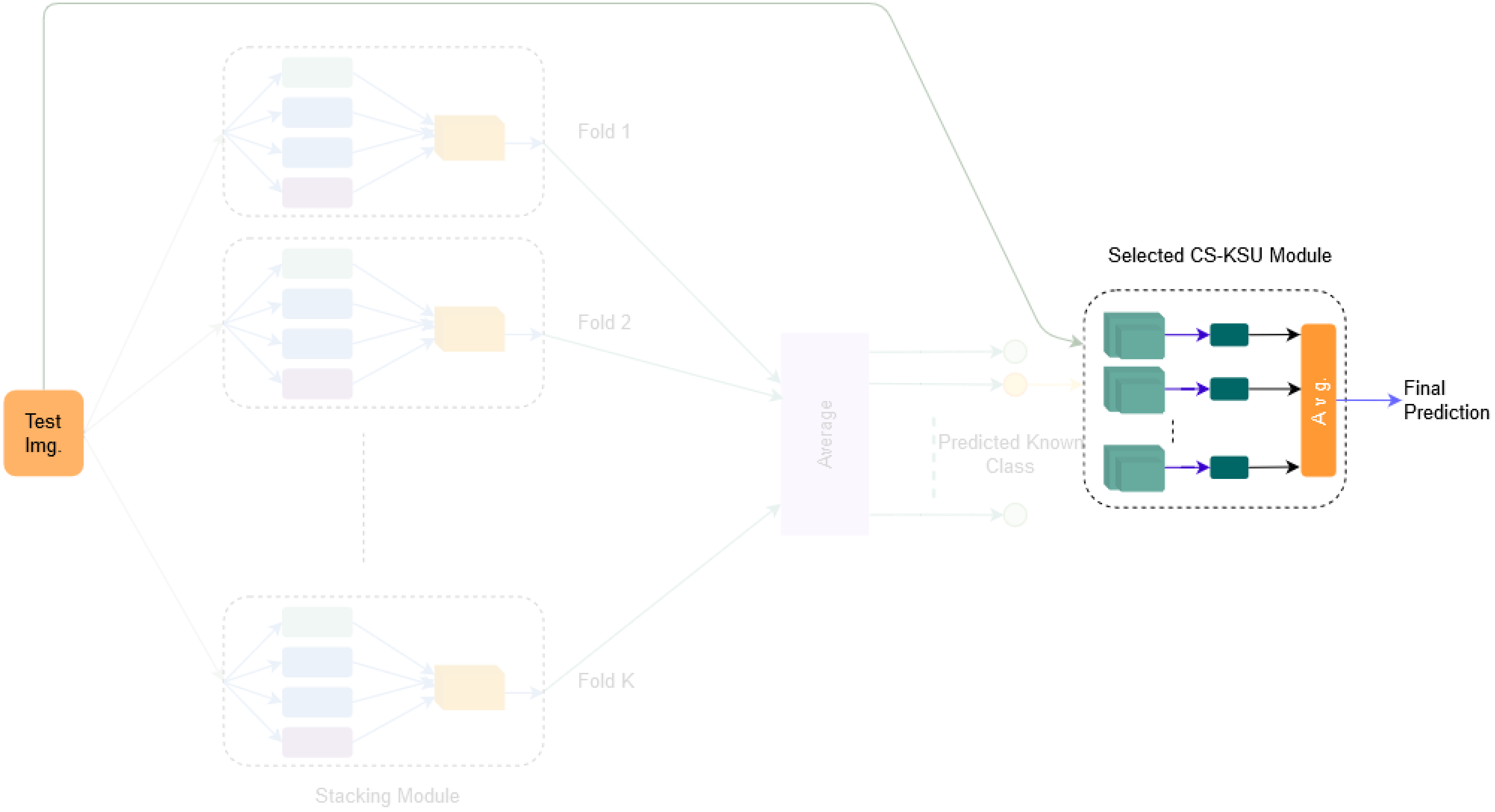
Testing Process – Explained



Testing Process – Explained



Testing Process – Explained



Results

Team/ Method	BMA	Unk. Class AUC	External Data
minjie (Ensemble)	0.632	0.705	Yes
Jost (Ensemble)	0.624	0.639	Yes
Sabanci University (Ensemble w/ ECOC)	0.602	0.582	No
Dermos (Ensemble)	0.595	0.500	No
Ours (Ensemble Avg. w/o Unknown detection	0.565	0.500	No
Ours Ensemble Stack w/o Unknown detection	0.591	0.500	No
Ours Ensemble Stack w/ Unknown detection	0.568	0.544	No

Table 1: Comparison with few other results from *ISIC 2019 Live Leaderboard*[1]

	Ensemble Avg.	Ensemble Stack	Ensemble Stack w/ Unk. Det.
MEL	0.825	0.825	0.801
NV	0.873	0.843	0.838
BCC	0.851	0.853	0.814
AK	0.698	0.777	0.757
BKL	0.752	0.742	0.675
DF	0.782	0.813	0.814
VASC	0.819	0.816	0.816
SCC	0.706	0.749	0.747
UNK	0.500	0.500	0.544
Avg. AUC	0.756	0.769	0.756

Table 2: Class-wise AUC[2] score of our different models

1. Our results stated, as compared on the ISIC Live Leaderboard 2019: Lesion Diagnosis *only*. URL: https://challenge2019.isic-archive.com/live-leaderboard.html
2. "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve", Hanley et. al. (1982)

ROC Plots

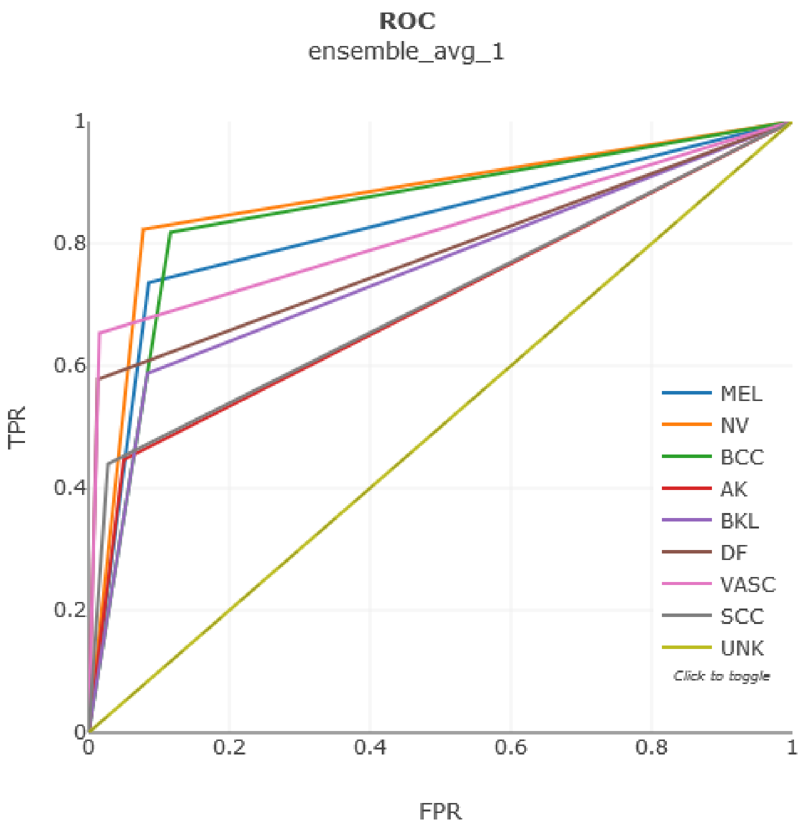


Figure: ROC plot for Average Model[1]

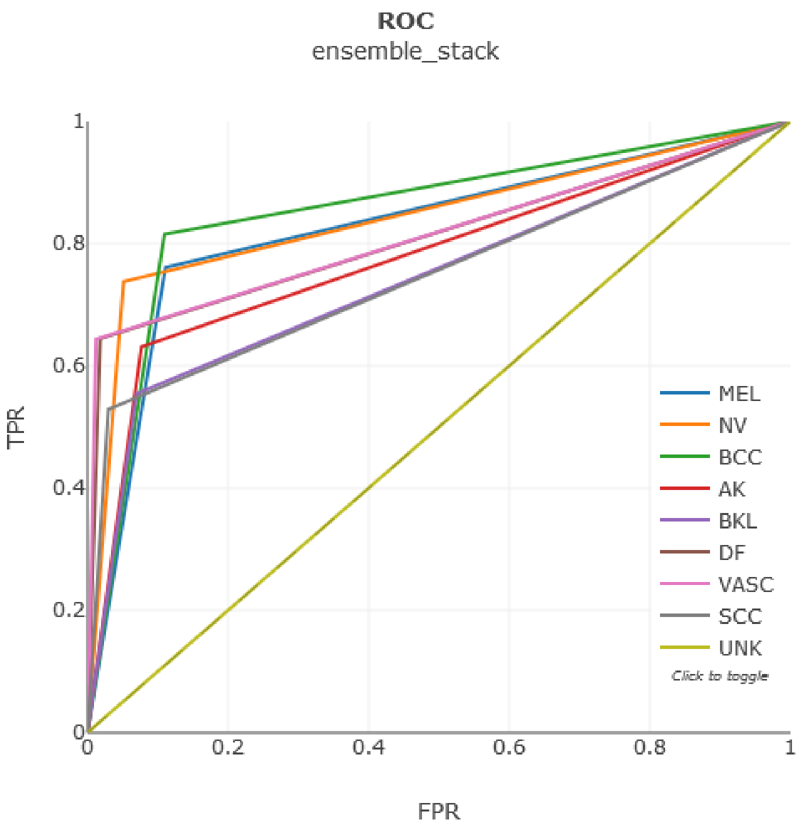


Figure: ROC plot for Stack Model[1]

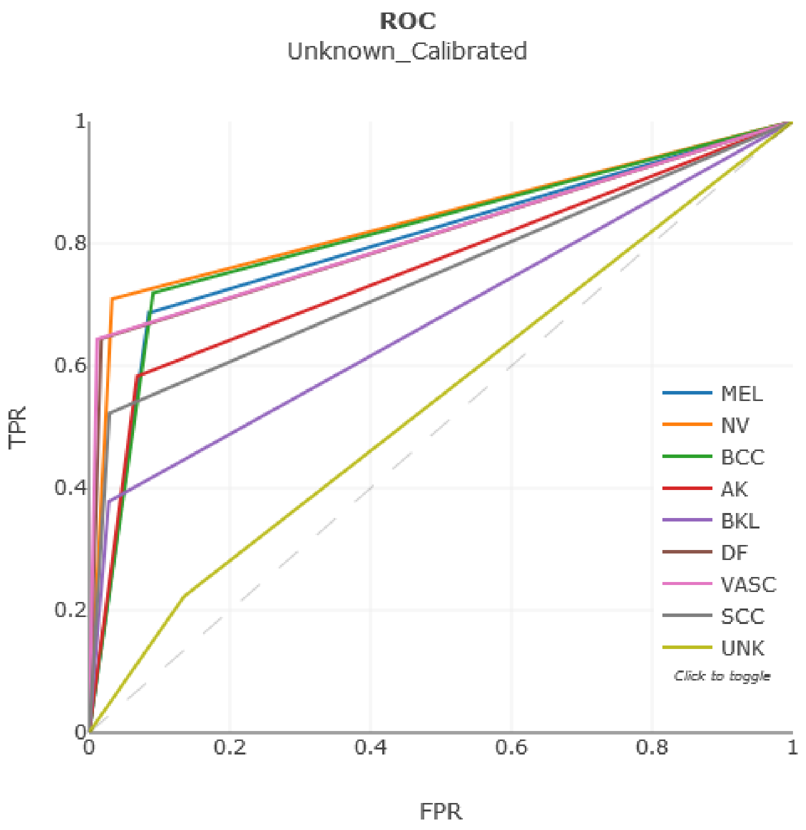


Figure: ROC plot for Stack plus CS-KSU Model[1]

1. Source ISIC Live Leaderboard 2019: Lesion Diagnosis. URL: https://challenge2019.isic-archive.com/live-leaderboard.html

Summary and Discussion

- A two-level hierarchical model was proposed in the work
- Stacking performs better than simple averaging, whereas CS-KSU module looks promising
- The hierarchical model is difficult to scale with increase in number of classes
- Trade off between AUC for Unknown class and BMA indicates the difficulty of the challenge
- The model’s performance may improve with extra data

Thank you!