EPIC: Egyptian Personal Image Classifier

By Ashraf Haress, Supervised by Prof. Nahla Barakat

Faculty of ICS,
Department of A.I,
The BUE

Abstract

Manually filtering Personal images (PIs) from irrelevant ones (IRIs) on one's phone gallery can be time consuming, thus a custom dataset was created from social media platforms like Facebook and Reddit, categorized to 9 classes, then trained on variants of CNN models where the hierarchical CNNs yielded the best average f1 score of 0.871.

Methodology

Model types used:

1. "m01" (VCNN, fig 1) which is a basic convolutional neural network for classification [1].
2. "m02" (MCNN, fig 2) used the same VCNN along with metadata input consisting of image's features (such as #faces in the image, etc).
3. "m03" (HCNN, fig 3), which uses multiple VCNNs to classify images in a hierarchical fashion; initially classifying if the image has high or low colour diversity, then classifies its flat class [2].
4. "m04" (XCNN, fig 4), which uses feature representations of images extracted from VCNN as input to XGBoost model.

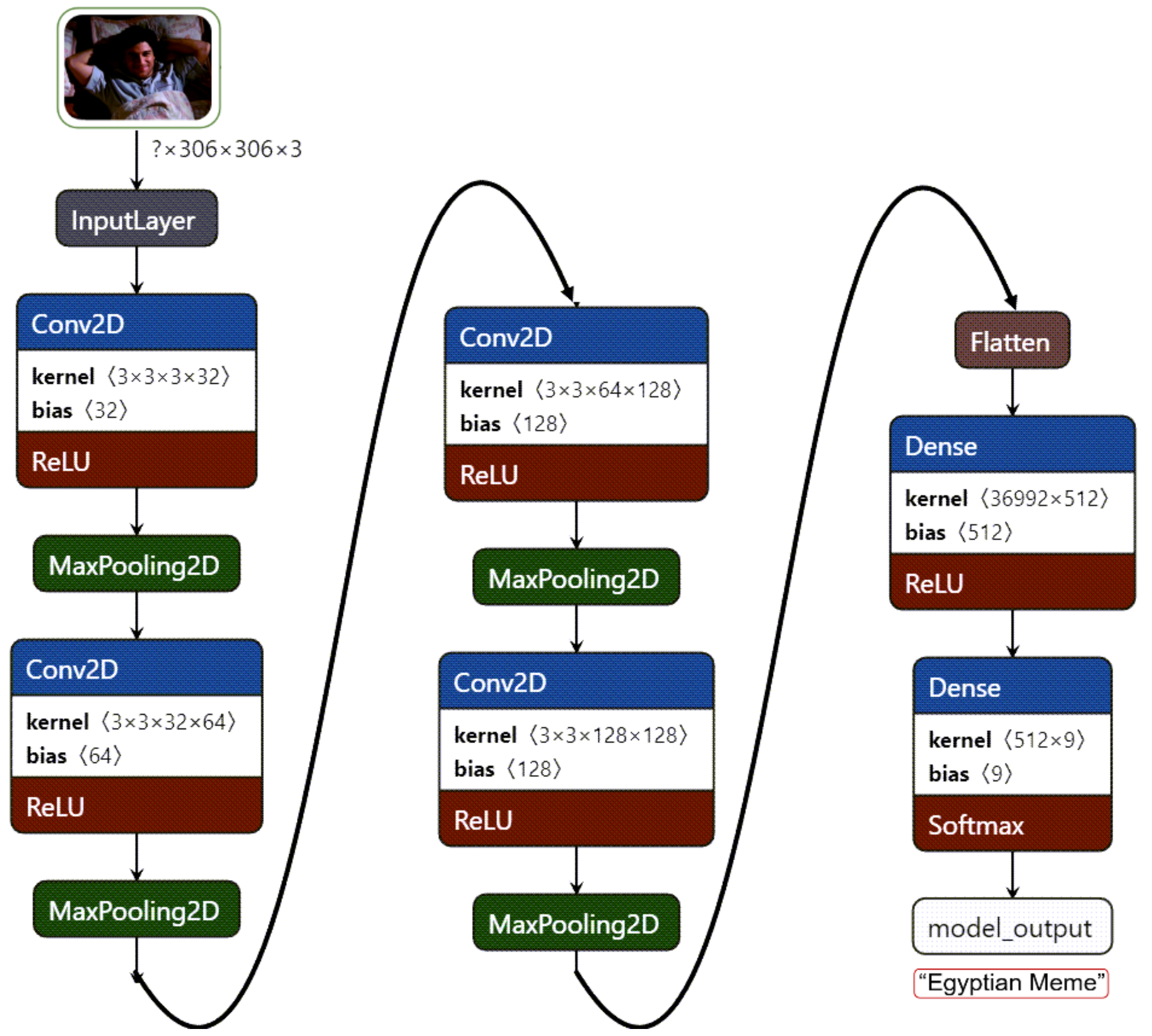
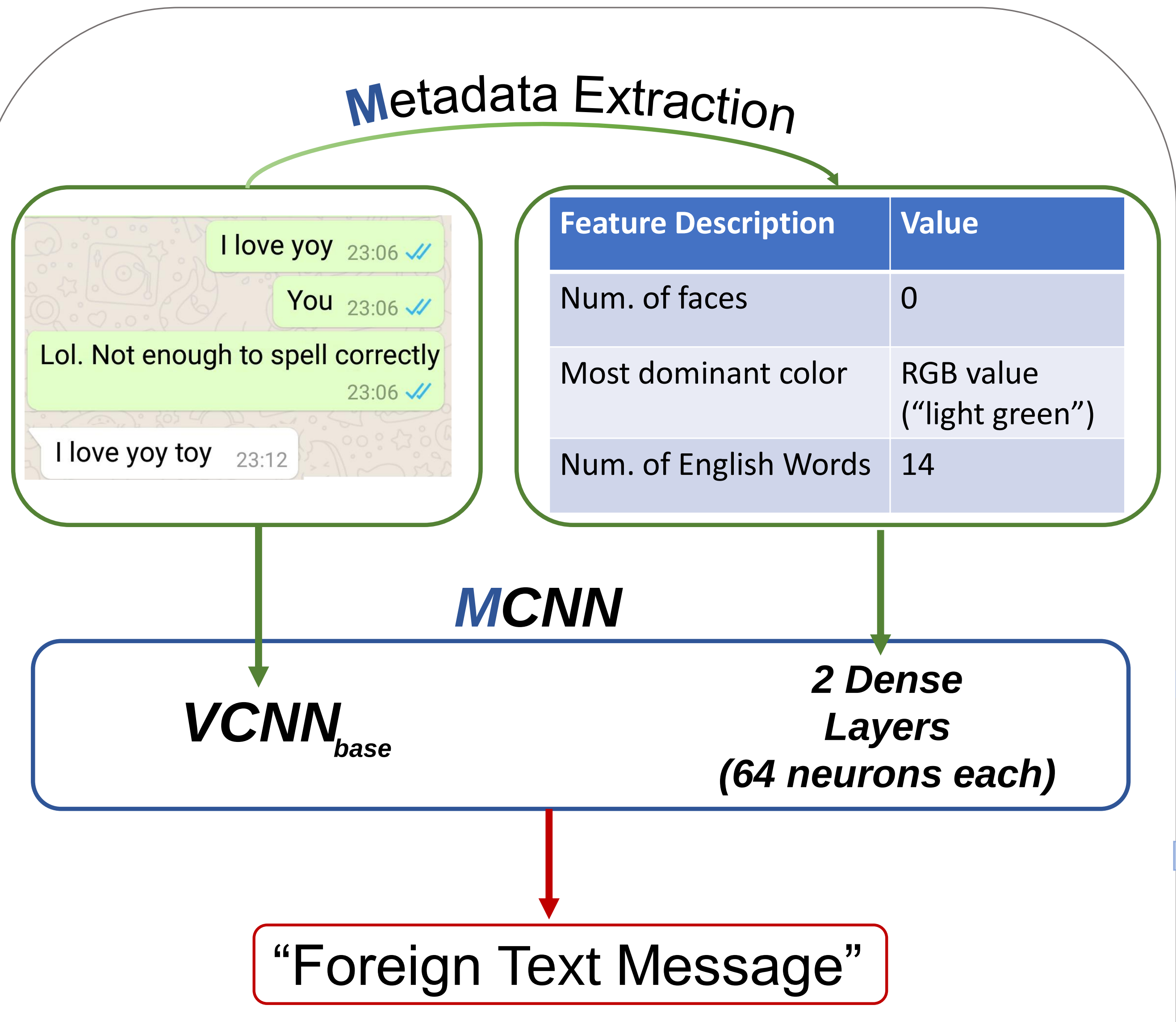


Fig 1. Vanilla CNN (VCNN$_{base}$) Architecture



I love yoy 23:06

You 23:06

Lol. Not enough to spell correctly 23:06

I love yoy toy 23:12

Feature Description	Value
Num. of faces	0
Most dominant color	RGB value ("light green")
Num. of English Words	14

Fig 2. MCNN Model Architecture

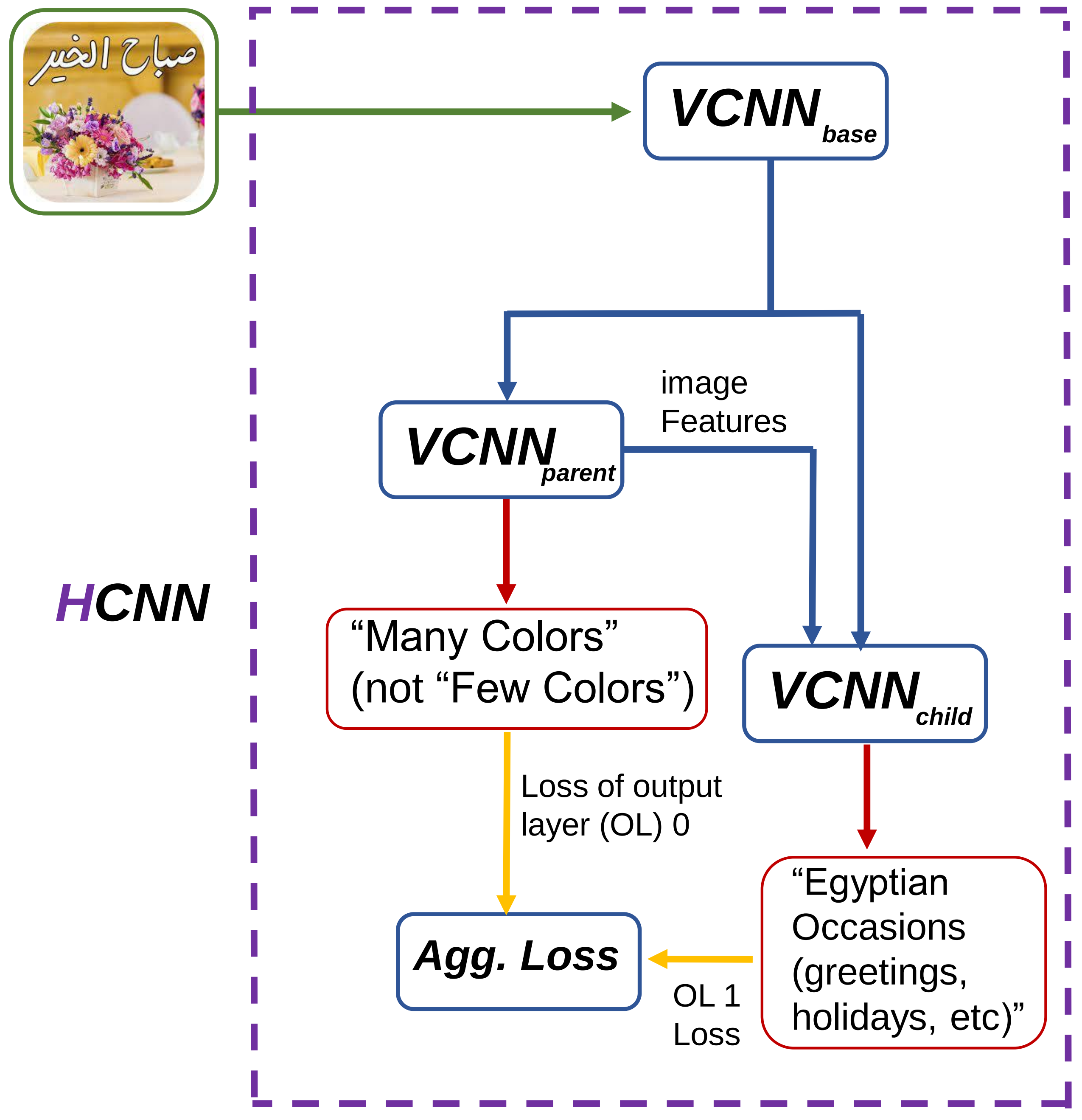

صباح الخير
HCNN
VCNN base
image Features
VCNN parent
"Many Colors" (not "Few Colors")
VCNN child
Loss of output layer (OL) 0
"Egyptian Occasions (greetings, holidays, etc)"
Agg. Loss
OL 1 Loss

Fig 3. Hierarchical CNN (HCNN) Model Architecture

VCNN$_{parent}$ has 512-128-64-2 dense layers

VCNN$_{child}$ has 512-128-64-9 dense layers

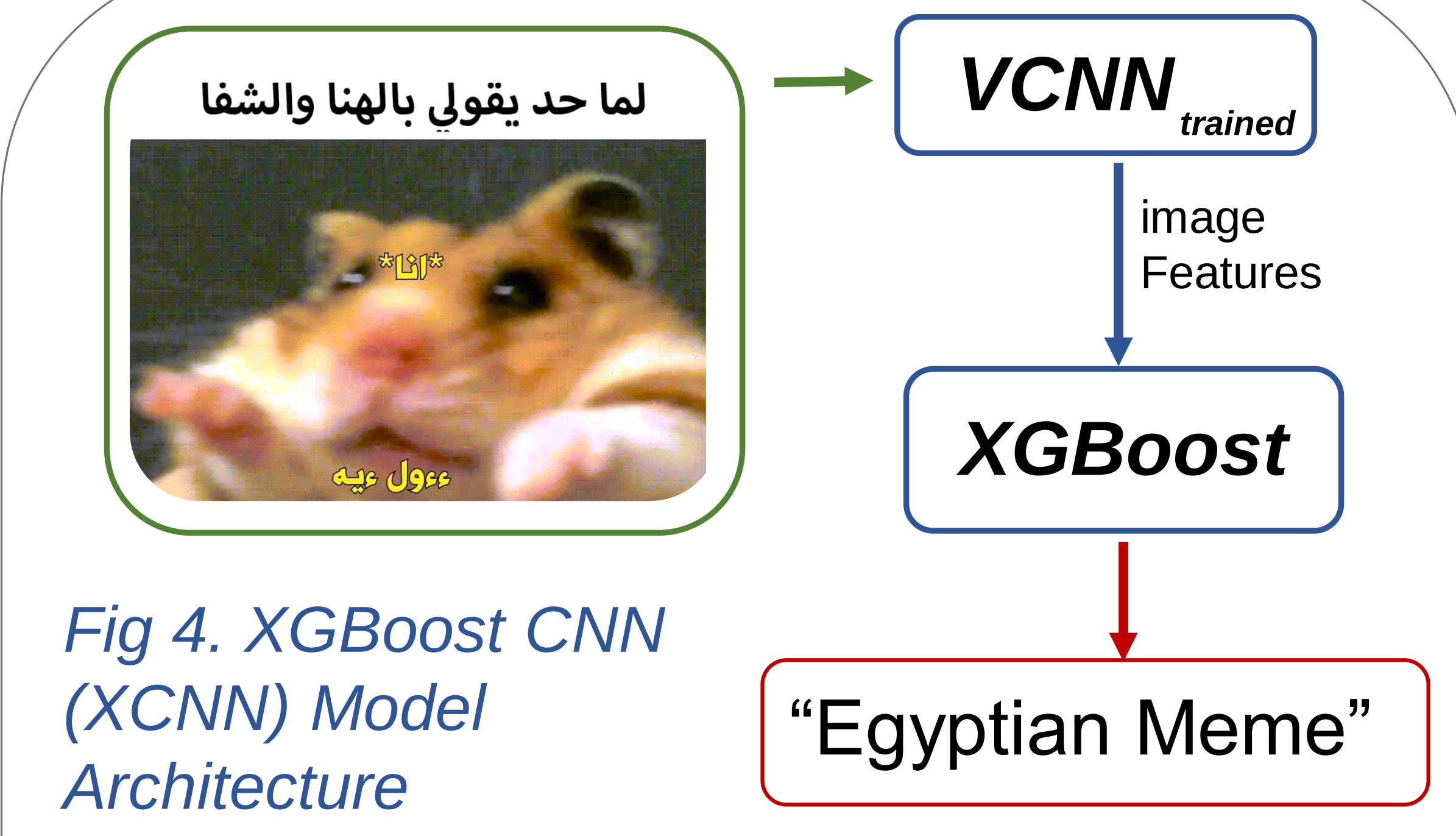


Fig 4. XGBoost CNN (XCNN) Model Architecture

Results

Qualitative results were visualised as input, output pairs in the previous pipeline figures. F1-score was used to measure different models (fig 5, 6), such that versions 01.x, 02.x, 03.x, 04.x correspond to VCNN, MCNN, XCNN, and HNN model variants, respectively, and described in Table 1.

Model Type	Model Version	Description
VCNN	01	no metadata, 8 batch size, 30 epochs, no augmentation.
	01.1	Same as m01, but with 32 batch size
	01.3	Same as m01, but with 16 batch size
	01.4	Same as m01.3, but with augmentation by rotating images
	01.6	Same as m01.1, but on dataset "v01.1" instead of "v01"
MCNN	02	color metadata, 16 batch size, 30 epochs, no augmentation
	02.1	Same as m02, but color metadata is scaled to [0, 1]
	02.2	Same as m02, but also with face metadata
	02.3	Same as m02.1, but also with face metadata
	02.5	Same as m02.1, but scaling face and text metadata as well
HCNN	03	1 level, stratified, no metadata, 32 batch size, 30 epochs, no augmentation
XCNN	04	no metadata, all batch, 100 epochs, no aug, no es, used m01.1
	04.1	Same as m04, but with sample weights
	04.2	Same as m04.1, but with Bayesian optimization and cross validation on train set

Table 1. Model Variants' Descriptions

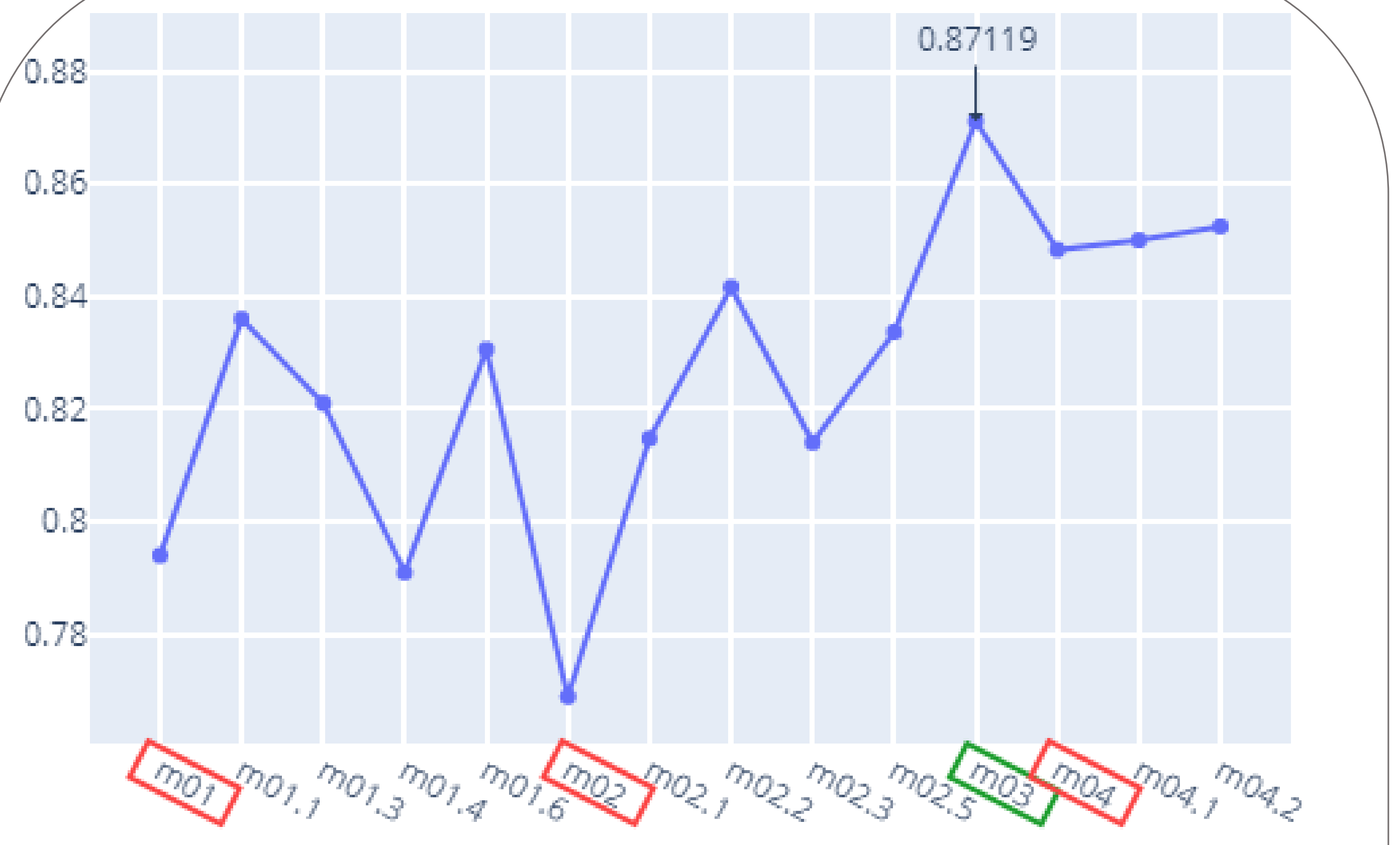
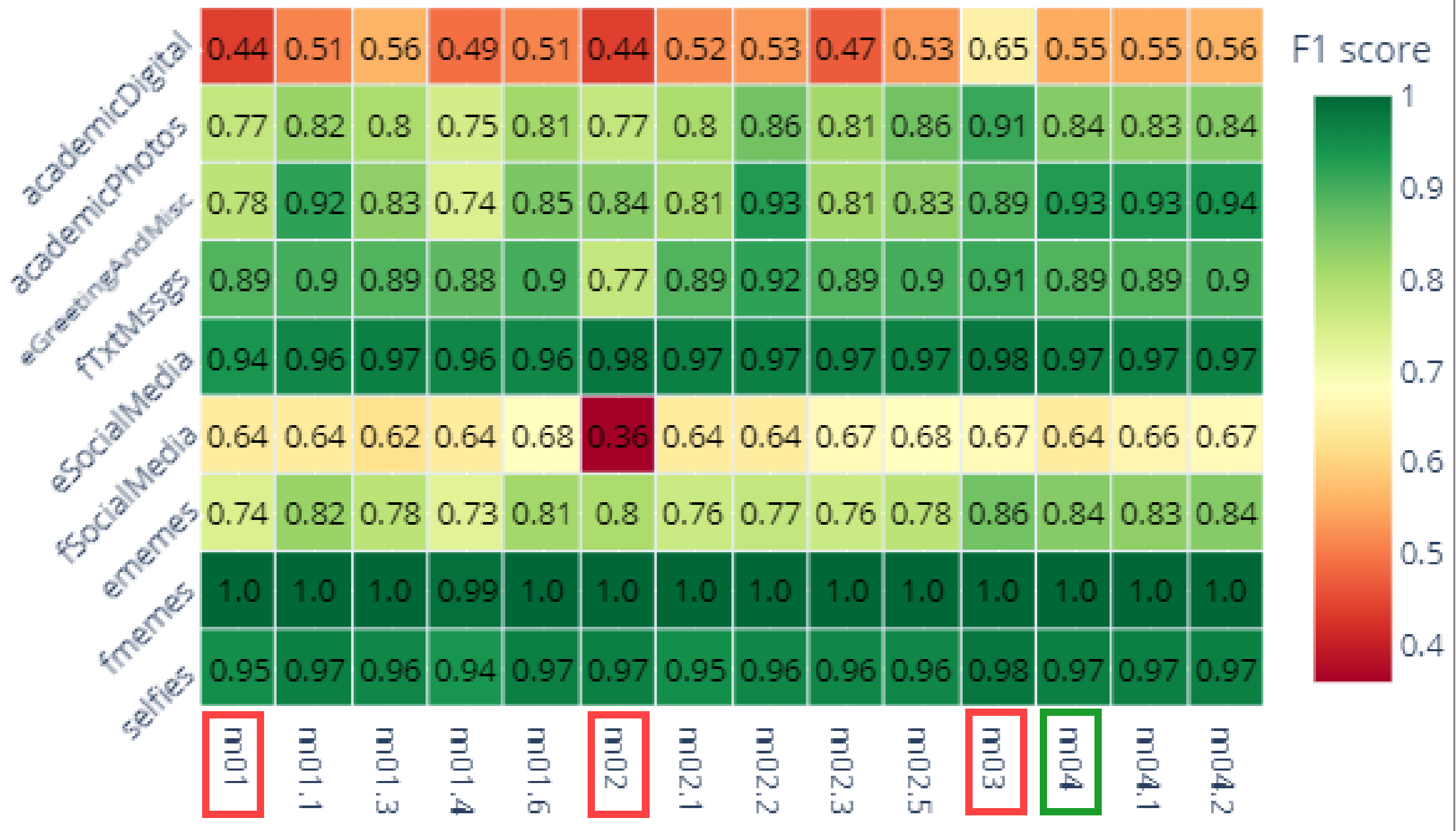


Fig 5. Average of Test F1-Scores for Each Model Version



	m01	m01.1	m01.3	m01.4	m01.6	m02	m02.1	m02.2	m02.3	m02.5	m03	m04	m04.1	m04.2
academicDigital	0.44	0.51	0.56	0.49	0.51	0.44	0.52	0.53	0.47	0.53	0.65	0.55	0.55	0.56
academicPhotos	0.77	0.82	0.8	0.75	0.81	0.77	0.8	0.86	0.81	0.86	0.91	0.84	0.83	0.84
eGreetingAndMisc	0.78	0.92	0.83	0.74	0.85	0.84	0.81	0.93	0.81	0.83	0.89	0.93	0.93	0.94
fTxtMssgs	0.89	0.9	0.89	0.88	0.9	0.77	0.89	0.92	0.89	0.9	0.91	0.89	0.89	0.9
eSocialMedia	0.94	0.96	0.97	0.96	0.96	0.98	0.97	0.97	0.97	0.97	0.98	0.97	0.97	0.97
fSocialMedia	0.64	0.64	0.62	0.64	0.68	0.36	0.64	0.64	0.67	0.68	0.67	0.64	0.66	0.67
ememes	0.74	0.82	0.78	0.73	0.81	0.8	0.76	0.77	0.76	0.78	0.86	0.84	0.83	0.84
fmemes	1.0	1.0	1.0	0.99	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
selfies	0.95	0.97	0.96	0.94	0.97	0.97	0.95	0.96	0.96	0.96	0.98	0.97	0.97	0.97

Fig 6. Test F1-Scores for Each Model Version

Conclusion

1. Increasing batch size is crucial for optimizing results (m01.1 > m01.3 >> m01)
2. Cleaning minority classes by removing images does not necessarily improve the score (m01.1 > m01.6)
3. Normalizing the input helps in improving results due it being a form model regularization (m02.1 > m02)
4. Breaking down a problem into simpler problems, i.e., hierarchy structure of HCNN, proves beneficial for the model. (m03 > all other models)

References

[1] J. Perez-Martin, B. Bustos, and M. Saldana, "Semantic Search of Memes on Twitter." arXiv, May 20, 2020. doi: 10.48550/arXiv.2002.01462.

[2] S. D. Das and S. Mandal, "Team Neuro at SemEval-2020 Task 8: Multi-Modal Fine Grain Emotion Classification of Memes using Multitask Learning." arXiv, May 21, 2020. Accessed: Nov. 23, 2022. [Online]. Available: http://arxiv.org/abs/2005.10915