Image Categorization Methodology using the IMFDB Dataset

Smita Darmora

1. Introduction

In the age of digital content and multimedia, precise and efficient categorization of images is crucial for a wide range of applications ranging from content recommendations to facial recognition systems. Recognizing faces in videos is a complex task because of variations in the pose, occlusion, blur, and resolution of captured images. However, this technology has multiple uses such as security monitoring and authentication. The "Image Categorization Methodology using the IMFDB Dataset" project aims to tackle this substantial challenge by utilizing machine learning and computer vision techniques [7] based on convolutional neural networks (CNNs).

The project's primary focus is on the Indian Movie Face Database (IMFDB) [9] [1], which contains 34,512 images exhibiting the faces of 100 Indian actors in over 100 movies. Each image within this extensive dataset encapsulates numerous factors, such as facial expressions, illumination conditions, pose variations, occlusions, age groups, makeup nuances, and gender distinctions. These attributes are intrinsic because of the manual selection and cropping process used during the extraction from movie video frames.

The project aims to categorize these images into seven primary categories, each of which further branches into multiple sub-classes. These primary categories encompass emotions (e.g., Anger, Happiness, Sadness), lighting conditions (e.g., Bad, Medium, High), facial poses (e.g., Frontal, Left, Right, Up, Down), occlusions (e.g., Glasses, Beard, Ornaments), age groups (e.g., Child, Young, Middle, Old), makeup details (e.g., Partial makeup, Over-makeup), and gender distinctions (Male, Female). The comprehensive representation of the image content in these categories and sub-classes provides an intricate and nuanced understanding of the underlying visual data. Table 1 sumarize the primary Categories and sub-classes.

The project employed a multifaceted approach to achieve image categorization. It utilizes multiclass classification models to classify images into multiple categories using convolution neural network (CNN). Additionally, pre-trained deep learning models were used, and transfer learning was applied to the features extracted from the CNN layers. This approach

Table 1: Summary of Primary Categories and Sub-classes

Primary Categories	**Sub-classes**
Expressions	Anger, Happiness, Sadness
Illumination	Bad, Medium, High
Pose	Frontal, Left, Right, Up, Down
Occlusions	Glasses, Beard, Ornaments
Age	Child, Young, Middle, Old
Makeup	Partial makeup, Over-makeup
Gender	Male, Female

significantly reduces the training time and enhances the prediction accuracy, particularly when dealing with limited data.

One notable strategy is optimizing the number of models required for each of its seven primary categories. Instead of utilizing seven separate models, a shared layer, and architectures capable of handling multiple categories simultaneously are explored which reduces the computational overhead.

The goal of this project is to enhance image classification using a combination of libraries including Pandas, Matplotlib, Seaborn, Keras, and TensorFlow. These libraries are instrumental in data management, model training, and the visualization of results. By combining these methodologies and utilizing the the IMFDB dataset, this project aims to improve image categorization using different CNN architecture.

2. Data Preparation

Data preparation and cleaning represent crucial initial steps. In my project, the following steps were meticulously followed.

2.1. Data sources and format

The primary data sources consisted of a CSV file containing metadata and path information regarding the images, and the actual image data were stored in .png format. Training and testing files were provided separately.

2.2. Data Cleaning with Pandas:

- To ensure data integrity and quality, a data-cleaning process was executed, primarily utilizing the Pandas library.
- This involved systematically inspecting the dataset for anomalies, such as missing data points or inconsistencies in column values.
- For instance, a thorough check was conducted to identify and rectify missing or erroneous data entries.

2.3. Data Categorization:

- To facilitate subsequent analysis and modeling, the data were organized into seven distinct categories aligned with the project's primary classification criteria.
- A crucial addition to the dataset is the inclusion of an additional column that stores the file paths in the corresponding image files.

2.4. Build a input pipeline:

Image data is typically large in size, and using it in its entirety for training a model can lead to memory issues. To address this, an input pipeline were constructed. Separate directories were established for training and testing purposes. Inside these directories, seven distinct folders for different models were created. Each of these model folders was then populated with its respective categories. For instance, in the case of the 'Age' model, directories are created such as /train/Age/male and /train/Age/female for training, with similar structure adopted for testing. Data was loaded using keras utility: `tf.keras.utils.image_dataset_from_directory`

Following these steps, the dataset was carefully prepared and ensured to be free of any potential issues that could impact the image categorization process. Adhering to these rigorous data preparation practices established a solid basis for the accurate analysis and classification of the IMFDB dataset.

3. Build the Model (Convolutional Neural Network)

As proposed, seven multi-class classification models using Convolutional Neural Networks (CNNs) were build. These models are designed to classify inputs into one of several classes, playing an instrumental role in predicting the various subclasses associated with each primary category. For each model, three different strategies were used. First, to built a base CNN model and then, to improve the accuracy, two different approaches were applied: model architecture with batch normalization and spatial dropout, and transfer tearning using ResNet50. For this document, one particular model is discussed in detail, as similar approaches were used for all other models. Considering there are seven primary categories to predict, our approach focuses on exploring strategies to minimize the number of necessary models.

3.1. Base CNN Architecture:

In this model, a convolutional Neural Network (CNN) for classification were employed, structured to process images of size 224 x 224 pixels. The model architecture consists of three convolutional layers, each followed by a max pooling layer. The first two convolutional layers have 32 filters of size 3 x 3 while the third layer increases the complexity with 64 filters. Following the convolutional layers, the output were flattened to feed it into a dense layer of 64 neurons, all employing ReLU activation functions. Importantly, the final dense layer is designed with four neurons, corresponding to the four unique age categories identified in our dataset. This model is compiled using the Adam optimizer, with a loss function of Sparse Categorical crossentropy to handle multi-class classification. The training approach involves 10 epochs, utilizing a batch size of 32, and the model's performance is validated on a separate validation dataset. Figure 8 shows the architecture of the model. For the visual representation of the different CNN architectures, Netron [2] was used, which is a viewer for neural network, deep learning, and machine learning models.

3.2. CNN Architecture with Batch Normalization and Spatial Dropout:

In this architecture, Sequential Convolutional Neural Network with (CNN) with batch normalization [8] and spatial dropout [6] were used. This model initiates with a 2D convolutional layer with 32 filters of size 3x3 and ReLU activation, tailored for input images of size 224x224 pixels with 3 color channels. Following this, a batch normalization layer is applied to stabilize and accelerate the training. After each of the first two convolutional layers, a max pooling layer was included with a 2x2 pool size to reduce the spatial dimensions of the output. Additionally, after the second pooling layer, a spatial dropout layer with a dropout rate of 50% were incorporated to prevent overfitting by randomly omitting a portion of the feature detectors. The network then progresses through another convolutional layer with 64 filters, followed by batch normalization and max pooling, and finally flattens the output for dense layer processing. A dense layer with 64 neurons employing ReLU activation and L2 regularization (lambda = 0.001) is used, followed by batch normalization, before concluding with a final dense layer representing the unique age categories. For optimization, Stochastic Gradient Descent (SGD) optimizer were used with an initial learning rate of 0.01 and a momentum of 0.9. The model is compiled using Sparse Categorical crossentropy as the loss function, targeting accuracy as the performance metric. To enhance the training process, early Stopping and reduce learning rate on plateau was implemented. The model is trained on a dataset with validation data included and both callbacks active. Figure 9 shows the architecture of the model.

3.3. CNN Architecture leveraging Transfer Learning:

In this architecture a classification model leveraging the powerful ResNet50 [5] architecture, pre-trained on ImageNet data was used. This model, accommodate input images of size 224x224 pixels, is utilized without its top layer. It begans by freezing the layers of the base ResNet50 model to preserve the pre-trained features, ensuring that only the newly added layers are trainable during the learning process. On top of this base model, a new architecture comprising a flatten layer to convert the 2D feature maps to a 1D vector was constructed, followed by a dense layer with 64 neurons employing ReLU activation and L2 regularization (with lambda = 0.001) for added complexity and to mitigate overfitting. This is further accompanied by batch normalization and a dropout layer with a rate of 50% to enhance generalization. The final output layer is a dense layer with a number of neurons equal to the unique categories in our dataset, using the softmax activation function for multi-class classification. For optimization, the stochastic gradient descent (SGD) optimizer were employed with an initial learning rate of 0.01 and momentum set to 0.9. The model compilation was done using sparse categorical crossentropy as the loss function, with a focus on accuracy as the primary metric. To refine the training process, early stopping was implemented which monitor the validation loss and adapt the learning process accordingly, enhancing efficiency and preventing overfitting. The training was conducted on a designated dataset, with validation data included for performance assessment. Figure 10 shows the architecture of the model.

3.4. Optimizing Model Count:

Considering there are seven primary categories for prediction, one might initially think that seven separate models are required. However, to optimize efficiency, a strategy were developed to reduce the number of necessary models. For this purpose,the 'Age' and 'Gender' models were merged into a single model named 'Gender_Age,' which predicts both the gender and age of the actor. Similarly, the 'Makeup' and 'Illumination' models were combined to form a new model called 'Makeup_Illumination' that predicts the actor's makeup category and the illumination of the picture.

In the categorization, we define four age groups: Child, Young, Middle, and Old; and two gender categories: Male and Female. After merging these categories in the new model 'Age_Gender', seven distinct categories: Male_Child, Male_Young, Male_Middle, Male_Old, Female_Young, Female_Middle, and Female_Old were identified. Notably, there was no data available for the 'Female_Child' combination. Similarly, in the case of makeup, we have two categories: Partial and Over. For Illumination, there are three categories: Bad, Medium, and High. Combining these in the new model 'Makeup_Illumination' results in six combinations: Partial_Bad, Partial_Medium, Partial_High, Over_Bad, Over_Medium, and Over_High.

- **Age_Gender:** Male_Child, Male_Young, Male_Middle, Male_Old, Female_Young, Female_Middle, and Female_Old
- **Makeup_Illumination:** Partial_Bad, Partial_Medium, Partial_High, Over_Bad, Over_Medium, and Over_High

After developing two new models, three distinct strategies were employed by building a base Convolutional Neural Network (CNN) model. To enhance accuracy, we integrated Model Architecture enhancements, including Batch Normalization and Spatial Dropout. Additionally, transfer learning with ResNet50 were utilized, as previously discussed. The reason for creating only two new models was the lack of additional combinations that held significant meaning

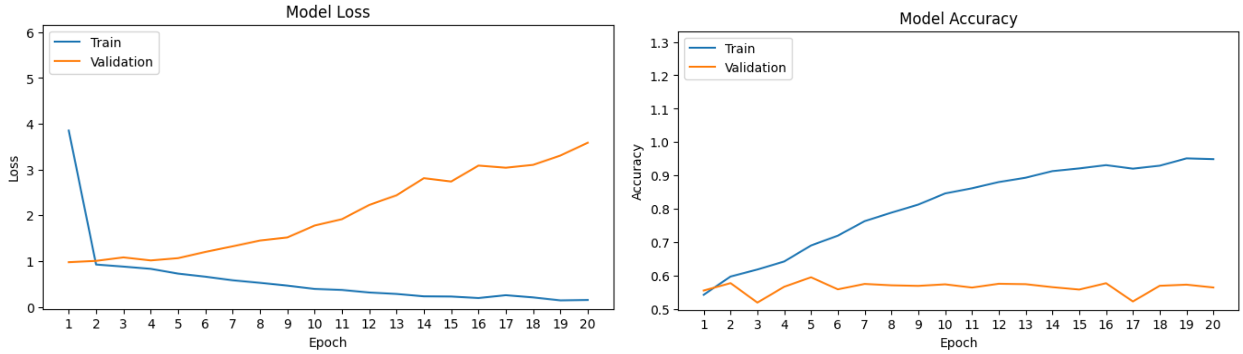
4. Results and Discussion:

This section is divided in two parts: Visualize training results and predict on new data. There are nine models to discuss, which include seven primary models and two combination models. For this report, I will discuss the 'Age' model in detail and only present the results for the other models.

4.1. Visualize training results

The first part of the discussion focuses on the visualization of training results. This crucial aspect of our CNN project involves analyzing and interpreting the performance metrics throughout the training process. By examining key indicators such as accuracy, loss, and validation metrics over successive epochs, we gain valuable insights into the model's learning trajectory and convergence behavior. Training / validation loss and accuracy was showed in Figure 1, 2 and 3 for base CNN model, CNN Architecture with Batch Normalization and Spatial Dropout and CNN Architecture leveraging Transfer Learning.

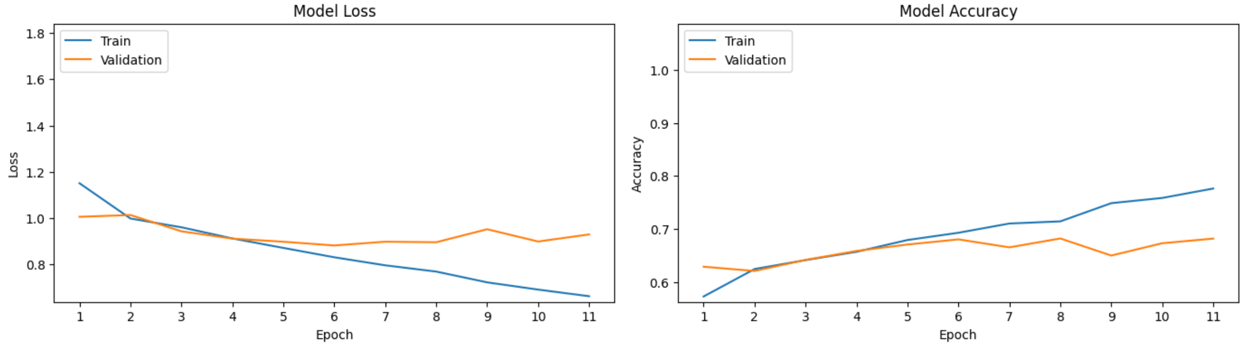
For the base model, the training accuracy of the 'Age' model was approximately 94%, while the validation accuracy was around 56%. However, after applying Batch Normalization and Spatial Dropout, there was a decrease in the training accuracy to 78%, but an improvement in the validation accuracy to approximately 68%. Additionally, after applying Transfer Learning using ResNet50, the training accuracy was observed to be around 95%.



(a) Training / Validation loss

(b) Training / Validation accuracy

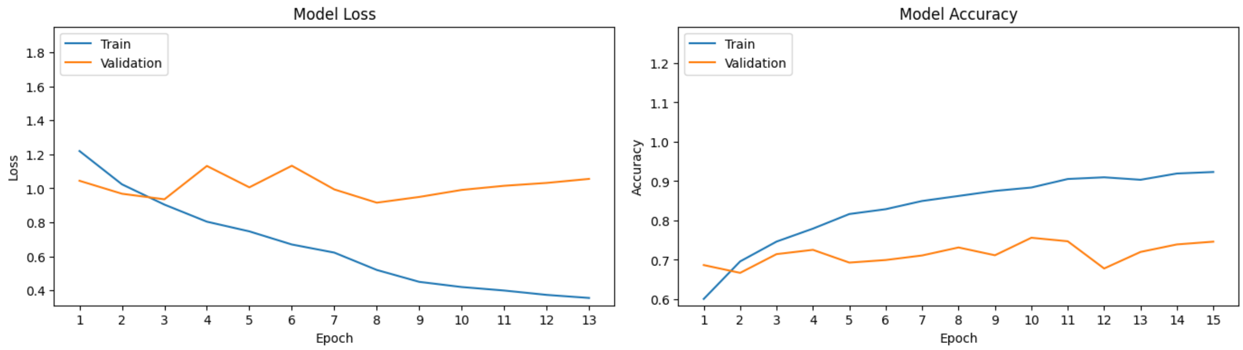
Figure 1: for Base CNN model



(a) Training / Validation loss

(b) Training / Validation accuracy

Figure 2: for CNN Architecture with Batch Normalization and Spatial Dropout



(a) Training / Validation loss

(b) Training / Validation accuracy

Figure 3: for CNN Architecture leveraging Transfer Learning

4.2. Predict on new data:

In this section of the report, the results of deploying the Convolutional Neural Network (CNN) model on previously unseen data are presented. This phase is crucial for evaluating the real-world efficacy of the model, demonstrating its ability to generalize beyond the data it was trained on.

For all nine models, the training, validation, and testing accuracy numbers are presented in the model performance Table 2. The testing results for one model, ”Age” are discussed here. The figures illustrates the training, validation, and testing accuracy for the ’Age’ model across three different CNN architectures. Figure 4 provides a visual representation of the presented data for ”Age” model. The testing accuracy for the base CNN model is ≈ 57%. After applying batch normalization and spatial dropout, the testing accuracy increased to ≈ 69%. However, when employed transfer learning with ResNet50, there is a notable improvement in the validation accuracy to about 76%. Figure 6 and Figure 7 showed all visual representation of all other models.

For transfer learning, I initially experimented with two well-recognized CNN architectures: VGG and ResNet50. These architectures were applied to two distinct models: ’Pose’ and ’Expression’. In the comparative analysis, ResNet50 demonstrated superior performance compared to VGG, in accuracy as can be seen in Figure 5. Therefore, I decided to implement ResNet50 across all models.

Table 2: Model Performance Summary

	Basic CNN Model			Model Architecture with BN and SD			ResNet50 Model		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
GENDER	0.92	0.78	0.77	0.69	0.70	0.78	0.95	0.90	0.90
MAKEUP	0.97	0.96	0.97	0.98	0.97	0.97	0.98	0.96	0.97
EXPRESSION	0.65	0.36	0.37	0.55	0.45	0.42	0.72	0.47	0.47
POSE	0.92	0.64	0.64	0.81	0.70	0.71	0.85	0.73	0.74
OCCLUSION	0.82	0.66	0.66	0.83	0.74	0.74	0.94	0.79	0.80
AGE	0.85	0.56	0.56	0.85	0.71	0.72	0.92	0.77	0.76
ILLUMINATION	0.82	0.62	0.63	0.74	0.69	0.71	0.85	0.71	0.72
AGE_GENDER	0.90	0.54	0.54	0.87	0.67	0.68	0.89	0.75	0.75
MAKEUP_ILLUMINATION	0.81	0.61	0.62	0.79	0.68	0.69	0.86	0.69	0.70

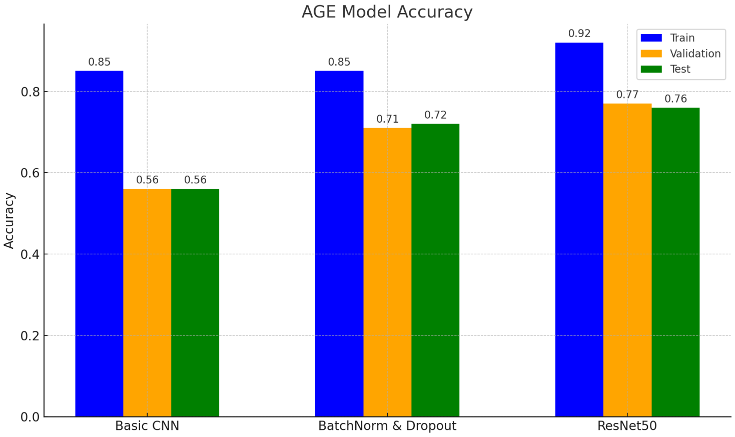


Figure 4: Comparison of three different CNN architecture

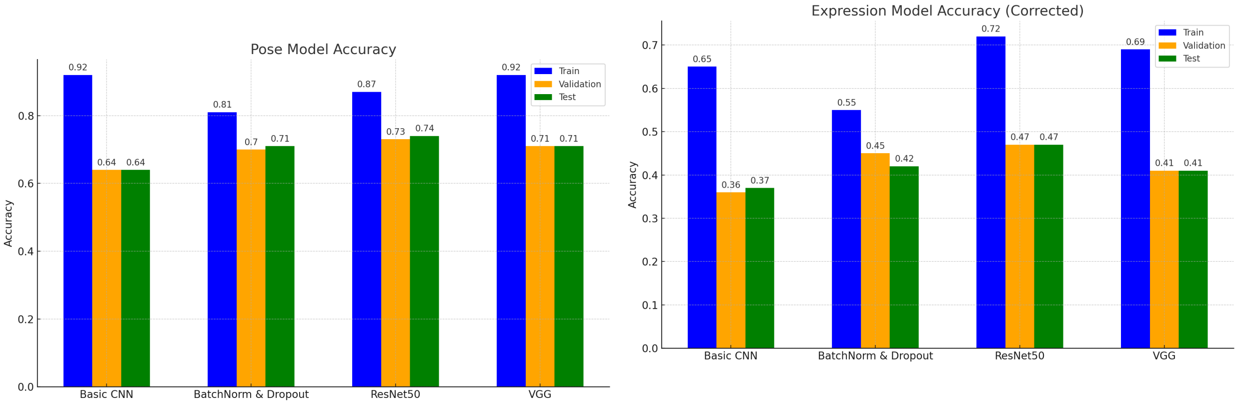


Figure 5: Comparison of ResNet50 and VGG

5. Conclusion

The study's objective is to utilize the Indian Movie Face Database (IMFDB) to develop an advanced image categorization system employing convolutional neural networks, aiming to classify images into seven primary categories. The project employs multiclass classification models and transfer learning to enhance efficiency and accuracy, as well as to optimize the model count. Following are the key lessons and insights I've gained from the project.

- In this study, I have gained a profound understanding of the importance of data comprehension, preparation, and cleaning before applying any model. This experience has highlighted that thorough data analysis and preprocessing are crucial steps in ensuring the effectiveness and accuracy of the models we develop.

- As expected with the basic CNN architecture, we observed overtraining, indicated by the substantially lower testing accuracy compared to the training accuracy.

- To mitigate overtraining and enhance accuracy, applying techniques such as batch normalization and spatial dropout significantly improved the testing accuracy across all models.

- Leveraging transfer learning further improved accuracy.
- When comparing VGG with ResNet50, ResNet50 outperformed, which aligns with previous reports [3] and [4].
- Combining different models can optimize the model count, but the accuracy of the combined models was less than that of the individual models.

For future scope, we observed that the accuracy on the test set for the combined models was lower than the accuracy of the individual models on their respective test sets. To address this, image augmentation techniques can be utilized. Additionally, there were some images that were not very clear in terms of certain labels; these images can be filtered out to improve the dataset's quality

References

[1] https://cvit.iiit.ac.in/projects/IMFDB/.

[2] https://github.com/lutzroeder/netron.

[3] R. Deepu R.C. Shivamurthy A. Victor Ikechukwu, S. Murali. Resnet-50 vs vgg-19 vs training from scratch: A comparative analysis of the segmentation and classification of pneumonia from chest x-ray images,. *Global Transitions Proceedings, Volume 2, Issue 2*, 2019. doi: https://doi.org/10.1016/j.gltp.2021.08.027.

[4] Viktar Atliha and Dmitrij Šešok. Comparison of vgg and resnet used as encoders for image captioning. *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–4, 2020. doi: 10.1109/eStream50540.2020.9108880.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2015.

[6] Lee C. Lee, S. Revisiting spatial dropout for regularizing convolutional neural networks. *Multimed Tools Appl 79, 34195–34207*, 2020. doi: https://doi.org/10.1007/s11042-020-09054-7.

[7] Rajeshwar Moghekar and Sachin Ahuja. Deep learning model for face recognition in unconstrained environment. *Journal of Computational and Theoretical Nanoscience*, 2019. doi: 10.1166/jctn.2019.8518.

[8] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift,. *CoRR*, abs/1502.03167, 2015. doi: https://doi.org/10.48550/arXiv.1502.03167.

[9] Parisa Beham Jyothi Gudavalli Menaka Kandasamy Radhesyam Vaddi Vidyagouri Hemadri J C Karure Raja Raju Rajan Vijay Kumar Shankar Setty, Moula Husain and C V Jawahar. Indian movie face database: A benchmark for face rcognition under wide variations. *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 2013.

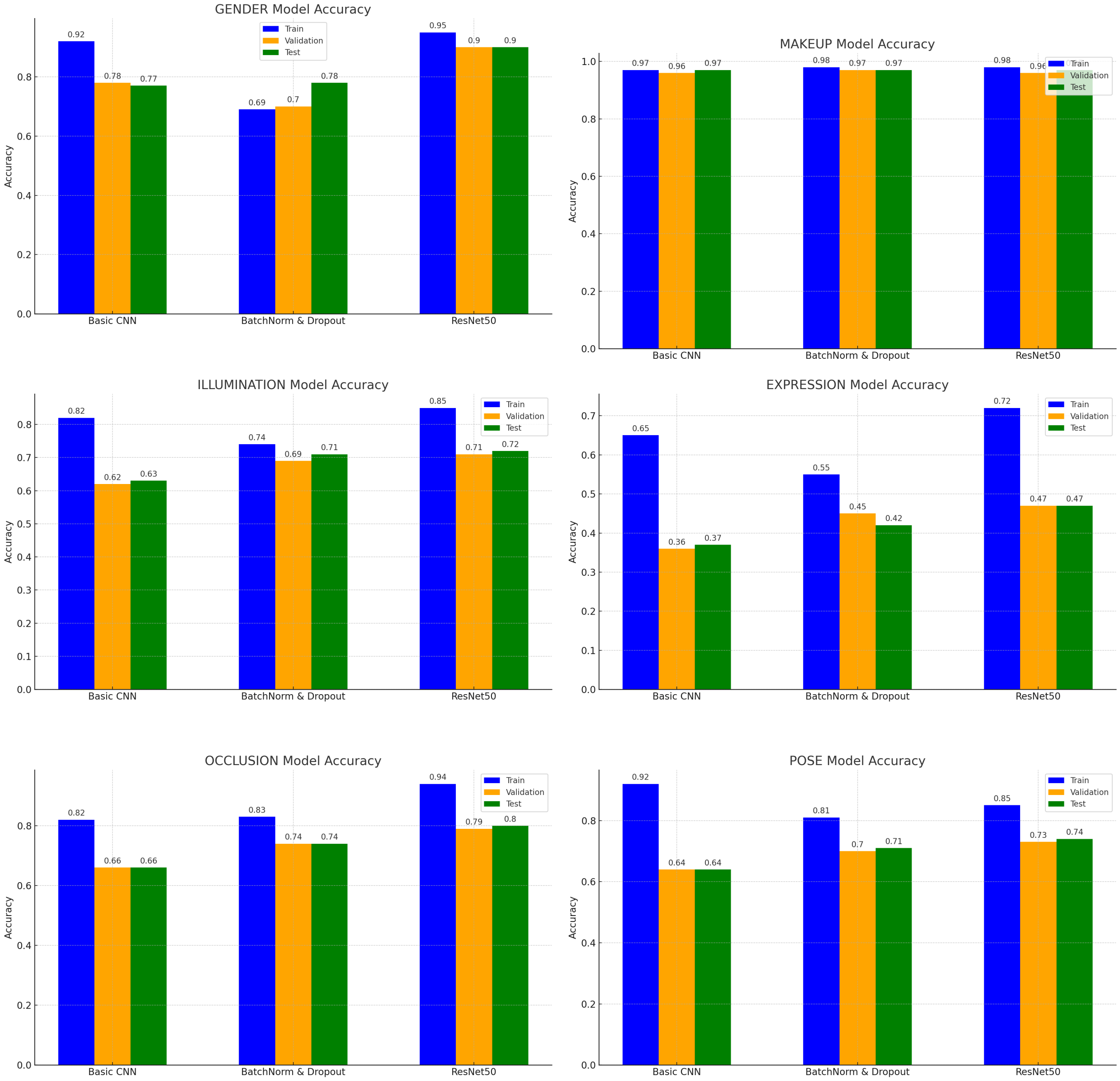


Figure 6: Comparison of three different CNN architecture for six different models

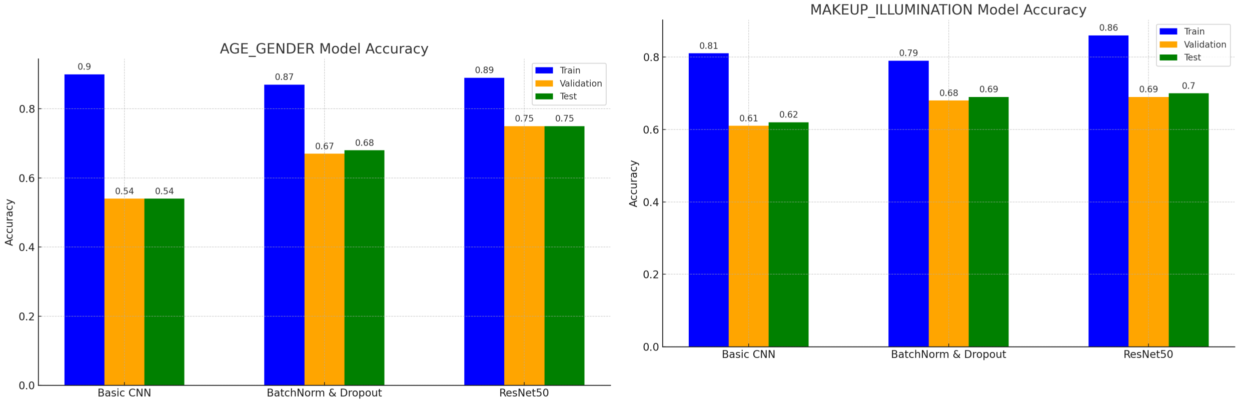


Figure 7: Comparison of three different CNN architecture for additional combined model

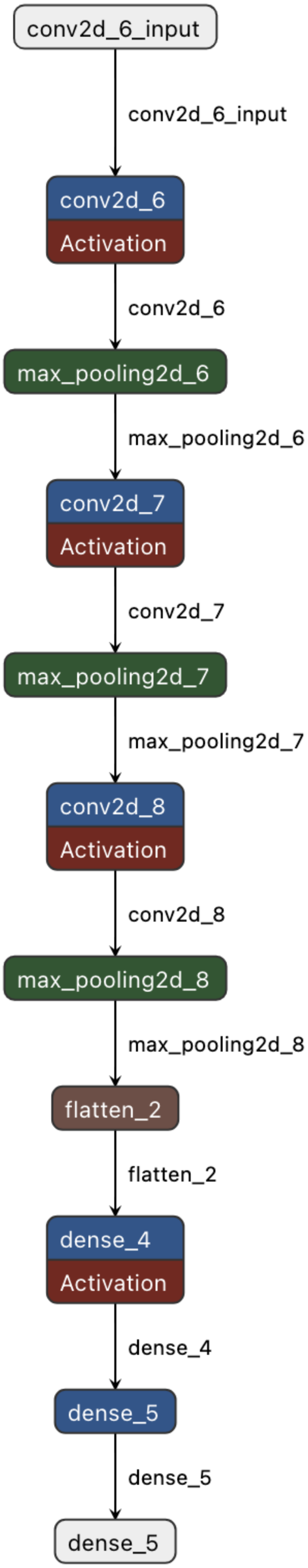


Figure 8: Base CNN model

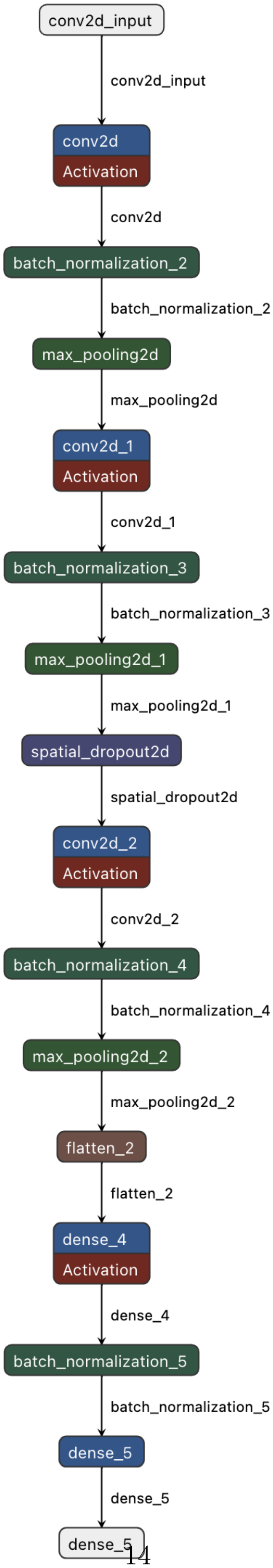


Figure 9: Batch Normalization and Spatial Dropout

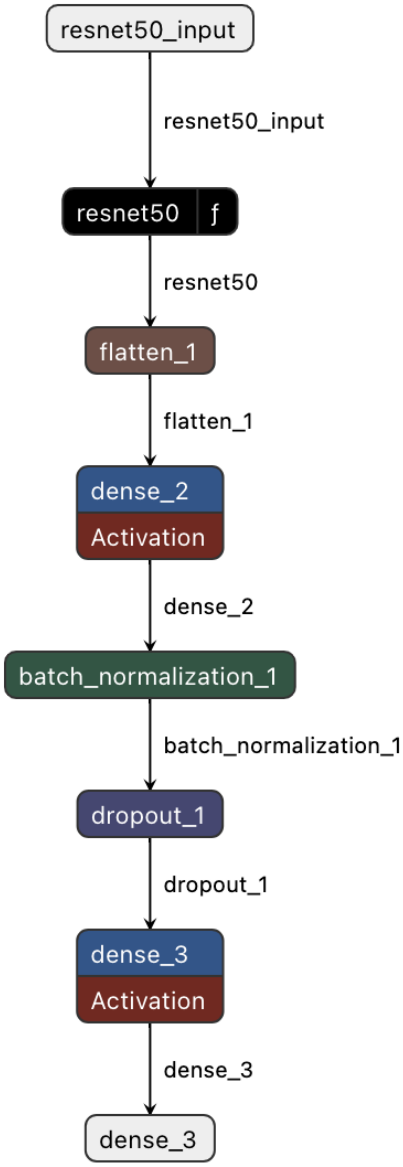


Figure 10: CNN Architecture leveraging Transfer Learning