

# Final Report - Image 6

names

## Executive Summary

Short description of the problem. The main findings. Key figure if appropriate. The practical relevance of the analysis.

## Aim and Background

A clear description of the problem, articulating the aim of this project. Provides appropriate multidisciplinary context and motivational background explained well in an appropriate language. Including background of the data

## Method - Data Collection & Developed Models

Clear description of the approach in data collection, developed model, the evaluation strategies from a data-science perspective. Here we refer to all types of evaluation metrics including graphical, qualitative and quantitative metric.

A clear description of your innovation and approach to the scientific problem.

What is your approach and which tools are used and why?

## Baseline Model

## Data Augmentation Models

## Method - Evaluation Strategies

## Results - Effect of Data Augmentation

A clear justification of the final approach based on the proposed evaluation strategies. Ensuring multiple evaluation strategies are used.

## Gaussian Noise

Observing Figure 1, the introduction of Gaussian noise in the implementation resulted in an increase in validation loss across all models, as compared to the original model. Notably, the application of random mean Gaussian noise yielded the largest validation loss for both the binary and categorical models. Moreover, a high mean Gaussian noise of 0.8 was applied led to the highest validation loss for both the binary and categorical models with class weights. Finally, the inclusion of low mean Gaussian noise demonstrated the highest validation loss for the RMSprop models, both with and without class weights.

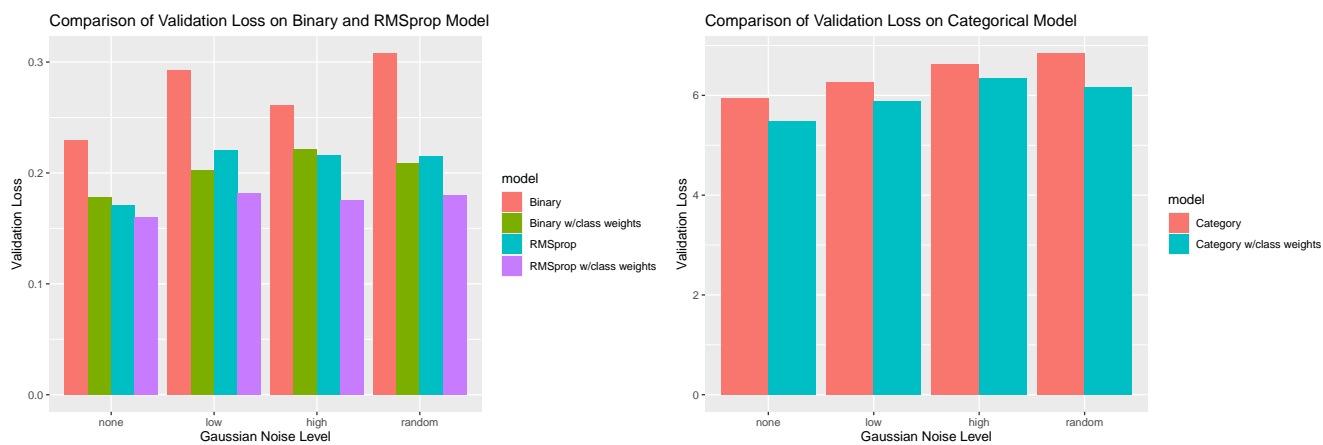


Figure 1: Validation Loss for Different Gaussian Noise Levels on CNN Models.

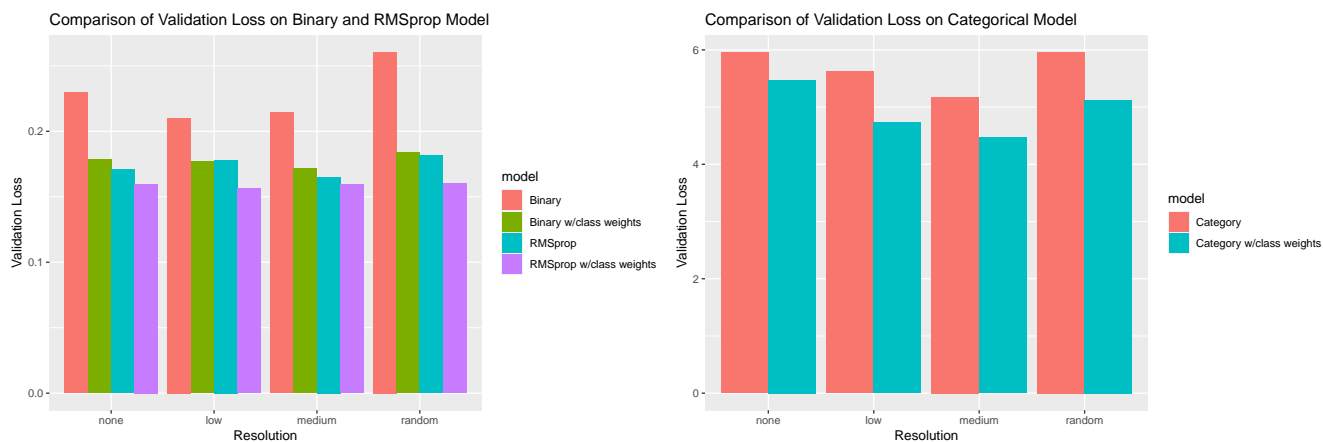


Figure 2: Validation Loss for Different Resolution on CNN Models.

## Resolution

The incorporation of low (16x16) and medium (32x32) resolutions in the training set resulted in a reduction of validation loss across all models, as illustrated in Figure 2. Notably, the categorical models with class weights experienced the most substantial benefits from this augmentation, with a decrease of 0.7503 when low resolution was added and a decrease of 1.0054 when medium resolution was utilized. Conversely, the introduction of random resolution into the training set led to an increase in validation loss, particularly noticeable in the binary model, with an increase of 0.0309.

## Rotation

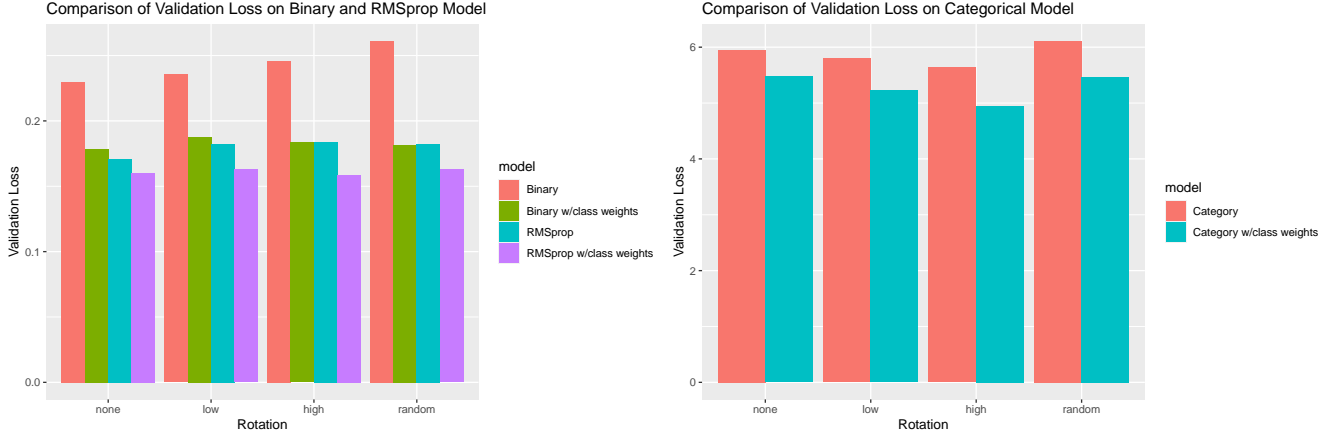


Figure 3: Validation Loss for Different Rotation on CNN Models.

While the binary and RMSprop models showed an increase in validation loss with rotation, the categorical models, both with and without class weights, demonstrated a decrease in validation loss when low (90 degrees) and high (180 degrees) rotations were applied, as shown in Figure 3. Models with random image rotation experienced the highest validation loss across all models.

## Combining Data Augmentations

The combination of all data augmentations resulted in a reduction of validation loss across all models, except for the RMSprop model without class weights, as indicated in the summary table presented in Figure 4. Additionally, Figure 5 highlights that the combined models achieved a flatter learning curve, indicating an improvement in mitigating overfitting of the dataset. Notably, the most significant improvement was observed in the models utilising categorical cross-entropy, both with and without class weights, with a respective decrease of 1.3457 (21.87%) and 0.8985 (16.29%) in validation loss.

As summarised in Figure 4, all models with class weights experiences a significant decrease in validation loss. This trend is particularly pronounced in the binary models, which experienced an average decrease of 0.05887 (23.10%) in validation loss. Similarly, models utilising the RMSprop optimizer without class weights display an average decrease of 0.05842 (23.04%) in validation loss. Moreover, models employing the RMSprop optimizer in conjunction with class weights demonstrate an average decrease of 0.0239 (12.45%) in validation loss.

## Results - Deployment

A clear description of the deployment process. An engaging and clear illustration of the product (games, shiny app, learning device etc) with a discussion of concepts from multiple disciplines.

Table Summary of Validation Loss for Every CNN Model

		Binary	Binary w/class weights	Cateogorical	Categorical w/class weight	RMSprop	RMSprop w/class weight
		Val Loss	Val Loss	Val Loss	Val Loss	Val Loss	Val Loss
No Noise		0.2298	0.185	6.1524	5.5162	0.1718	0.1599
Gaussian Noise	0.2	0.2930	0.2029	6.2574	5.8756	0.2202	0.1821
	0.8	0.2607	0.2210	6.6276	6.3434	0.2156	0.1751
	Random	0.3078	0.2086	6.8412	6.4580	0.2149	0.1800
Resolution	16x16	0.2101	0.1770	5.6171	4.7255	0.1782	0.1563
	32x32	0.2145	0.1714	5.1790	4.4704	0.1652	0.1592
	Random	0.2607	0.1840	5.9652	5.1211	0.1813	0.1602
Rotation	90	0.2355	0.1878	5.800	5.2267	0.1823	0.1628
	180	0.2460	0.1836	5.63470	4.9477	0.1840	0.1585
	Random	0.2611	0.1818	6.1100	5.4714	0.1824	0.1629
Combined Model		0.2084	0.1769	4.8067	4.6177	0.1891	0.1601

Figure 4: Summary Table of Validation Loss for All CNN Models.

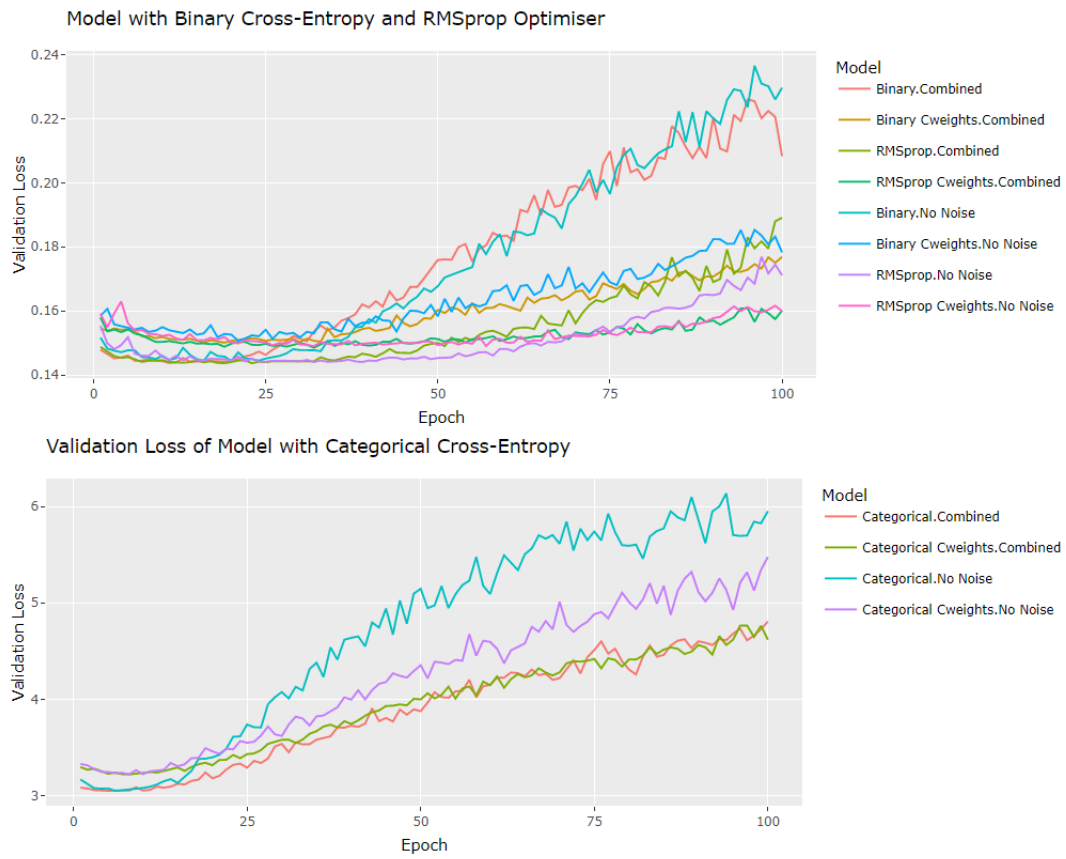


Figure 5: Learning Curves of Models with Combined Data Augmentation Compared to Models with No Noise

General results notes: Communicate the results to a general scientist, note this is different to the target audience of your product. What are the key findings? and use figures to illustrate the results.

All the plots above can be access through the **CCR Shiny App**.

## Discussion

How robust and generalizable is your finding or your product?

Discussion of potential shortcomings or issues associated with the development process or the product with reference to both disciplines.

Identification of future work or improvement in both disciplines.

## Conclusion

Conclusion adequately summarises the project and identification of future work.

**Student Contributions**

**References**

# Appendix

Main code & technical details of your approach