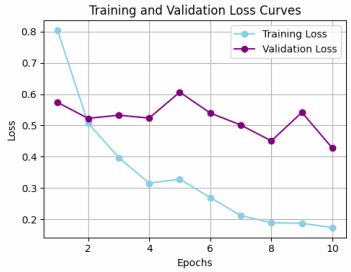
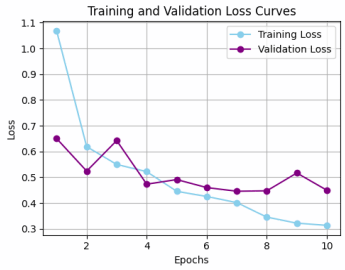
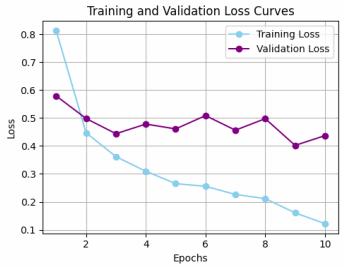
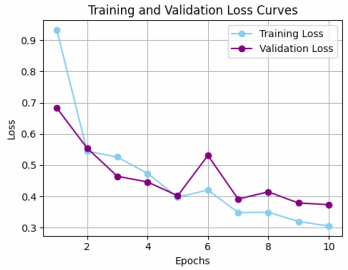
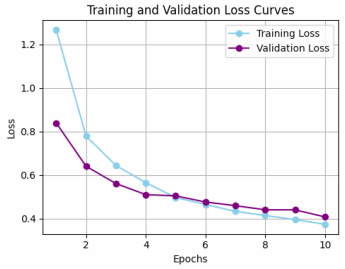
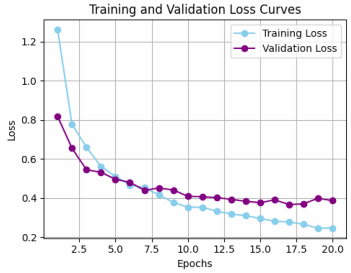
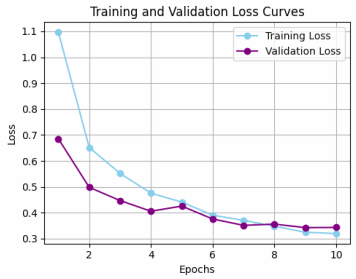
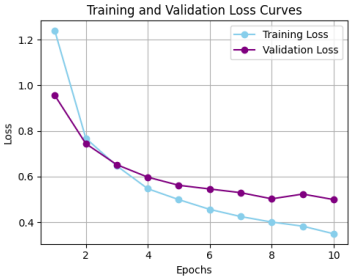
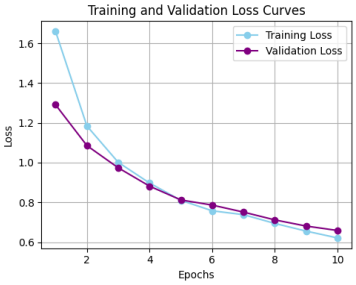
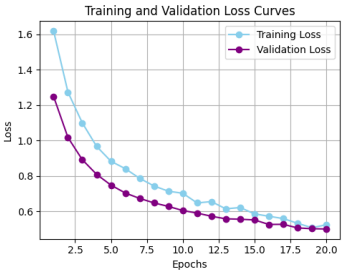
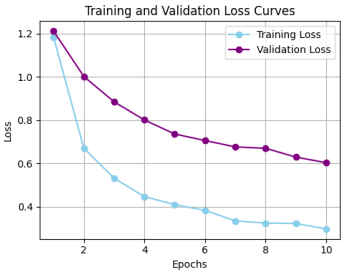
In order to improve this model, different parameters were tuned and evaluated to see its effect, and how the model performed.

Experimental Methods	Metrics	Validation		Test				
		Loss	Accuracy	Loss	Accuracy	Precision	Recall	F1 Score
1	**Before trash class oversampling (add 341 augmented trash images)** 	0.4274	0.8609	0.4362	0.8471	0.20	0.21	0.20
2	**Before trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x)* 	0.4495	0.8448	0.4930	0.8333	0.20	0.21	0.20
3	**After trash class oversampling (add 341 augmented trash images)** 	0.4371	0.8714	0.4262	0.8685	0.15	0.15	0.15
4	**After trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x)*	0.4195	0.8571	0.4186	0.8596	0.18	0.18	0.18

5	**After trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x) + Class weights* 	0.3738	0.8768	0.3665	0.8683	0.19	0.19	0.19
6	**After trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001)* 	0.4078	0.8518	0.4331	0.8458	0.18	0.19	0.19
7	**After trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001) + increase epoch no. from 10 to 20* 	0.3864	0.8786	0.3787	0.8683	0.19	0.19	0.19
8	**After trash class oversampling (add 341 augmented trash images) + Regularization with** *x = Dropout(0.5)(x) + Class weights*	0.3433	0.8899	0.3222	0.9013	0.16	0.16	0.16

	+ Adam(learning_rate=0.0001) + oversampling all 6 classes by 20% of their respective class size (after oversampling trash class) 							
9	**After class balancing, instead of the above trash class oversampling** (*oversampling the classes below median size to hit the median class size → so this obviously includes trash class*) **+ Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001)* **+ Corrected oversampling process** to only train set, instead of all 3 sets 	0.4996	0.8266	0.5210	0.8118	0.20	0.20	0.20
10	**After class balancing, instead of the above trash class oversampling** (*oversampling the classes below median size to hit the median class size → so this obviously includes trash class*) **+ Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001)* **+ Corrected oversampling process to** only train set, instead of all 3 sets + **Changed number of units (neurons) in a dense layer** from 1042 to 121 	0.6589	0.7823	0.6938	0.7647	0.20	0.20	0.19

11	**After class balancing, instead of the above trash class oversampling** (*oversampling 5 classes to the biggest class size = 594*) **+ Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001)* **+ Corrected oversampling process to** only train set, instead of all 3 sets + changed number of units (neurons) in a dense layer from 1042 to 121 	0.6032	0.7984	0.6938	0.7647	0.20	0.20	0.19
12	**Regularization with** *x = Dropout(0.5)(x) + Class weights + Adam(learning_rate=0.0001)* **+ Corrected oversampling process to** only train set, instead of all 3 sets + **changed number of units (neurons)** in a dense layer from 1042 to 121 + k-fold cross validation + **increase epochs** from 10 to 20 	0.5012	0.8376	0.2716	0.9549	0.20	0.20	0.20