## Template

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 1736 |  |
| Train set | 1389 |  |
| Val set | 227 |  |
| Test set | 69 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalizetion | [-1, 1] | [0, 1] / [-1, 1] |
| pre\_process | tekboart | base\_mode, tekboart |
| Dropout | 0.2 |  |
| L2 Regularization | None |  |
| Data Augmentation | YES (keras) |  |
| Class weight? | Yes |  |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=100 |  |
| #epochs | 20 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | EarlyStopping(patience=5, ‘val\_loss’) |  |
| lr\_rate (for fine-tune) | 0.001 / 100 |  |
| lr\_schedule\_fine-tune | No |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | No |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Performance Measure | Train | Val | Test |
| Loss |  |  |  |
| Accuracy |  |  |  |

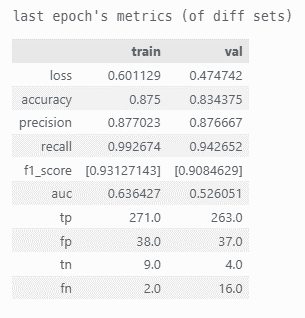
## ResNet50\_scratch (2022.08.23)

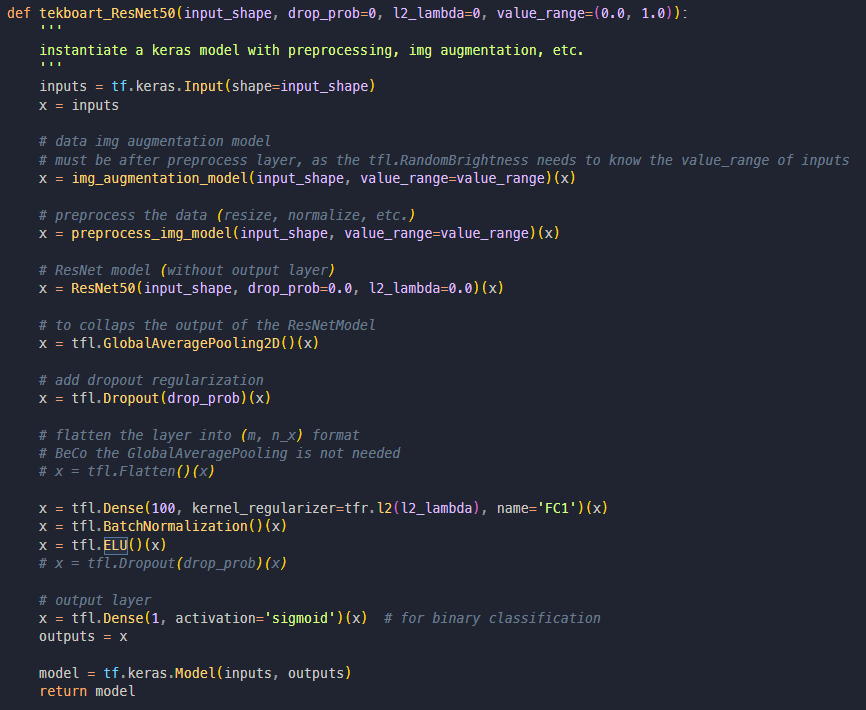
|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | tekboart\_ResNet50 | NO Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images |  |  |
| Train set |  |  |
| Val set |  |  |
| Test set |  |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | tekboart | base\_mode, tekboart |
| Dropout | 0.2 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | YES (keras) | only H-flip |
| Class weight? | Yes |  |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=100, stair\_case=T |  |
| #epochs | 100 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| Callback | NO |  |
| lr\_rate (for fine-tune) |  |  |
| lr\_schedule\_fine-tune |  |  |
| # un-freezed layers |  |  |
| #epochs (for fine-tune) |  |  |
| Callback (for fine-tune) |  |  |

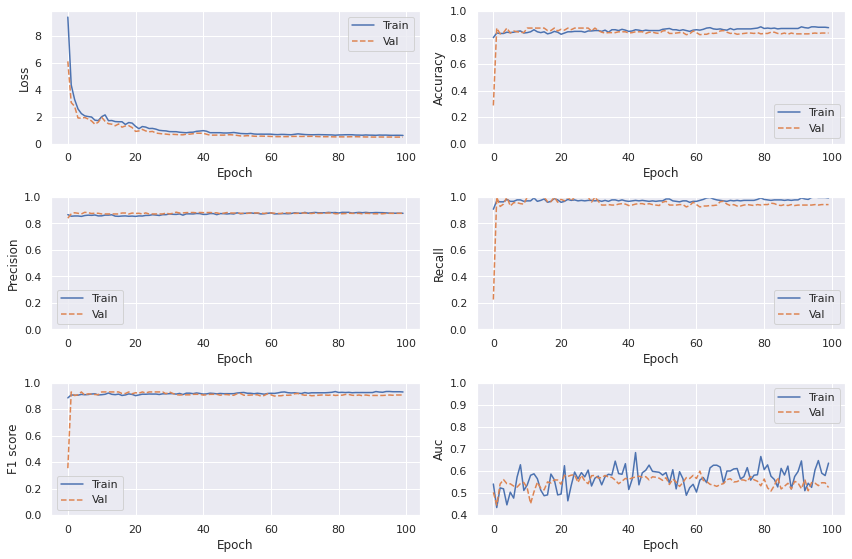
## ResNet50-v2\_scratch (2022.09.05) ⭐

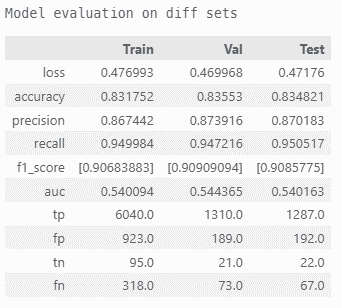
|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.8 |  |
| L2 Regularization | 0.1 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.01 |  |
| lr\_schedule | decay\_step=10, DR=.96, stair\_case=T | we used 10 BeC .take(10), or else 1,000 should have been used |
| #epochs | 100 |  |
| callback | - |  |
| Class weight? (for fine-tune) | - |  |
| lr\_rate (for fine-tune) | - |  |
| lr\_schedule\_fine-tune | - |  |
| # un-freezed layers | - |  |
| #epochs (for fine-tune) | - |  |
| Callback (for fine-tune) | - |  |

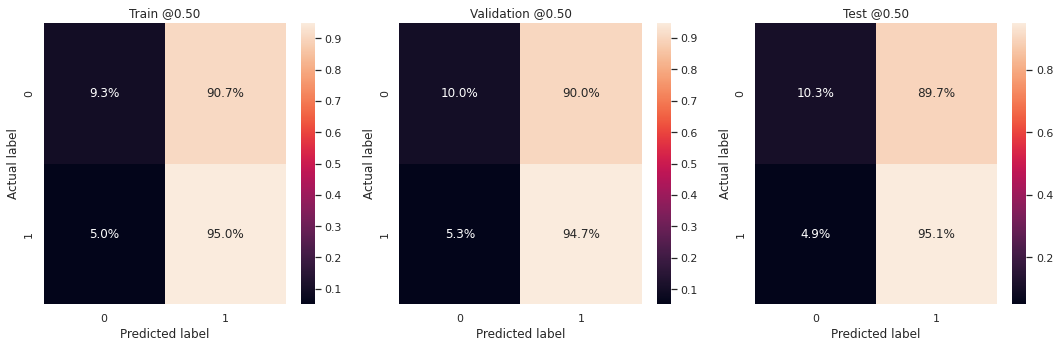
\* I used train\_ds.take(10) & val\_ds.take(10) for .fit() to get results faster (so becareful as these resutls are not completely relevant or correct)







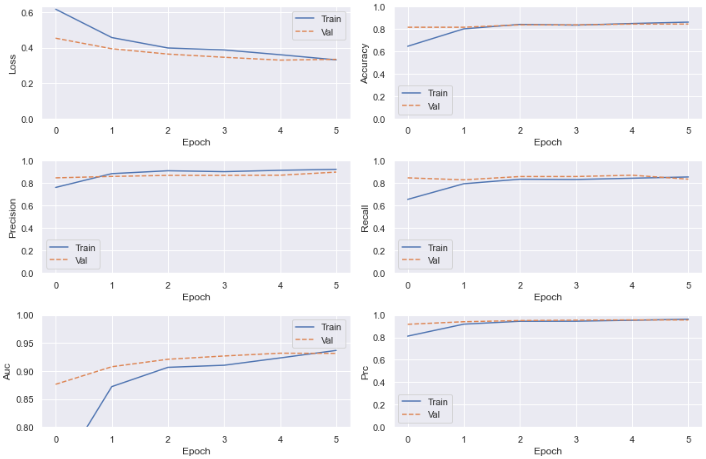




## ResNet50v2\_tl

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | ResNet50v2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 1736 |  |
| Train set | 1389 |  |
| Val set | 227 |  |
| Test set | 69 |  |
| batch\_size | 32 |  |
| input\_shape | (512, 512) | (64, 64), (150, 150), (224, 224) |
| normalizetion | [-1, 1] | [0, 1] / [-1, 1] |
| pre\_process | tekboart | base\_mode, tekboart |
| Dropout | 0.2 |  |
| L2 Regularization | None |  |
| Data Augmentation | YES (keras) |  |
| Class weight? | NO |  |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=100, decay\_rate=0.6, staircase=True |  |
| #epochs | 6 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | EarlyStopping(patience=5, ‘val\_prc’) |  |
| lr\_rate (for fine-tune) | 0.001 / 10 |  |
| lr\_schedule\_fine-tune | No |  |
| # un-freezed layers | +140 (50 out of 190) | Train layers from 140 and up |
| #epochs (for fine-tune) | 8 |  |
| Callback (for fine-tune) | No |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Performance Measure | Train | Val | Test |
| Loss |  |  |  |
| Accuracy |  |  |  |



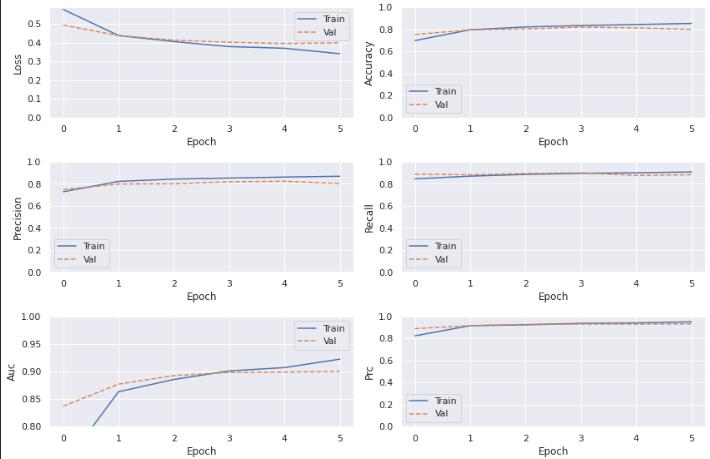
## ResNet50v2\_tl

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 1736 |  |
| Train set | 1389 |  |
| Val set | 227 |  |
| Test set | 69 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalizetion | [-1, 1] | [0, 1] / [-1, 1] |
| pre\_process | tekboart | base\_mode, tekboart |
| Dropout | NO |  |
| L2 Regularization | None |  |
| Data Augmentation | YES (keras) |  |
| Class weight? |  |  |
| lr\_rate |  |  |
| lr\_schedule |  |  |
| #epochs |  |  |
| Metrics |  |  |
| callback |  |  |
| lr\_rate (for fine-tune) |  |  |
| lr\_schedule\_fine-tune |  |  |
| # un-freezed layers |  | Train layers from 100 and up |
| #epochs (for fine-tune) |  |  |
| Callback (for fine-tune) |  |  |

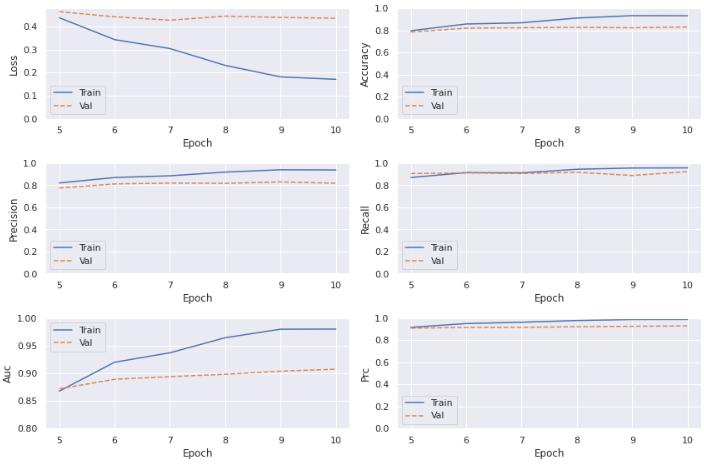
\* the ResNet50\_v2 worked a lot faster & better than the ResNet50 we built from scratch, which is not a surprise as we had only 1736 images in total, and we know that transfer learning works great with small data.

\* using the pretrained ResNet50v2 with [-1, 1] normalized inputs worked better than the [0, 1] values, so it’s safe to assume that ResNet50v2 had been trained with [-1, 1] values, rather than [0, 1].

\* using 224x224 images, in comparison with (150x150) ones, reduced overfitting and caused the model (i.e., ResNetV2\_pretrained) to perform a lot better



\* train only our added binary output (Dense(1))

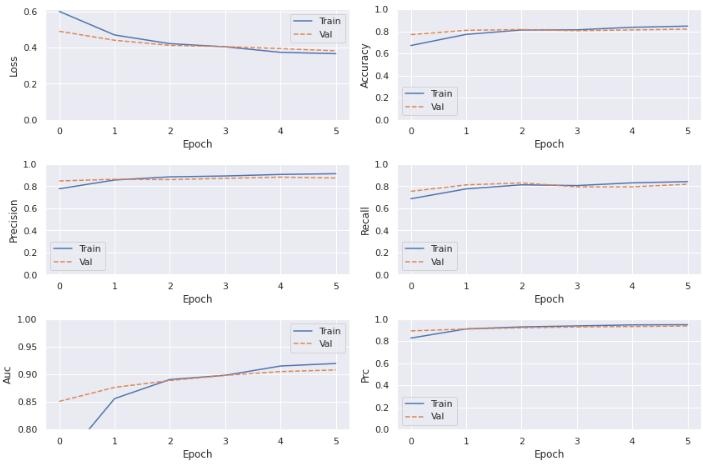


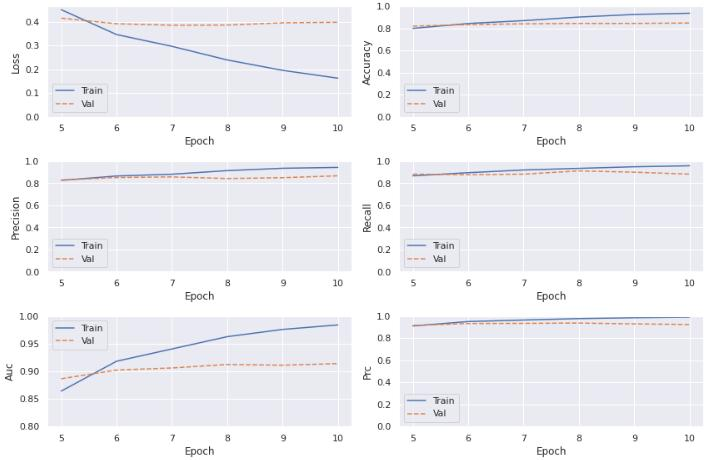


\* after fine-tune (by unlocking 50 of top layers )

**Results when using weight\_class (as our data is imbalanced) (no dropout)**

\* the result are better when applying the weight\_class, but not sure whether it’s the result of randomness in model (e.g., data augmentation,

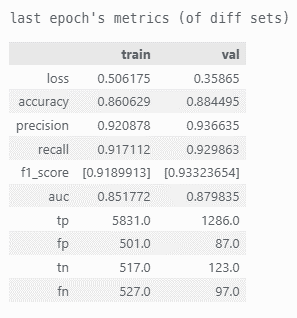






## ResNet150-v2 (2022.09.05) ⭐

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.7 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 10 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2, 1: 1} |
| lr\_rate (for fine-tune) | 1e-8 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +500 (64 out of 564) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





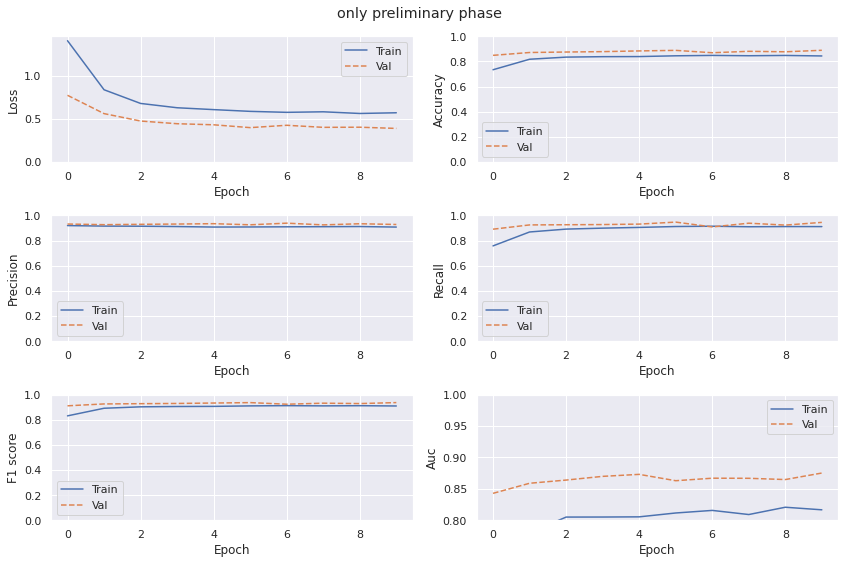


Figure 1: train the self added layers (i.e., Dense(1))

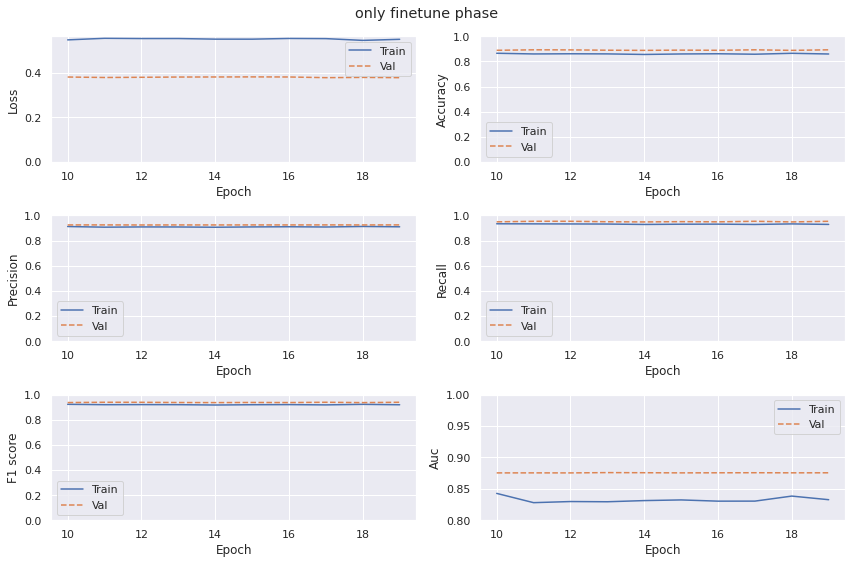
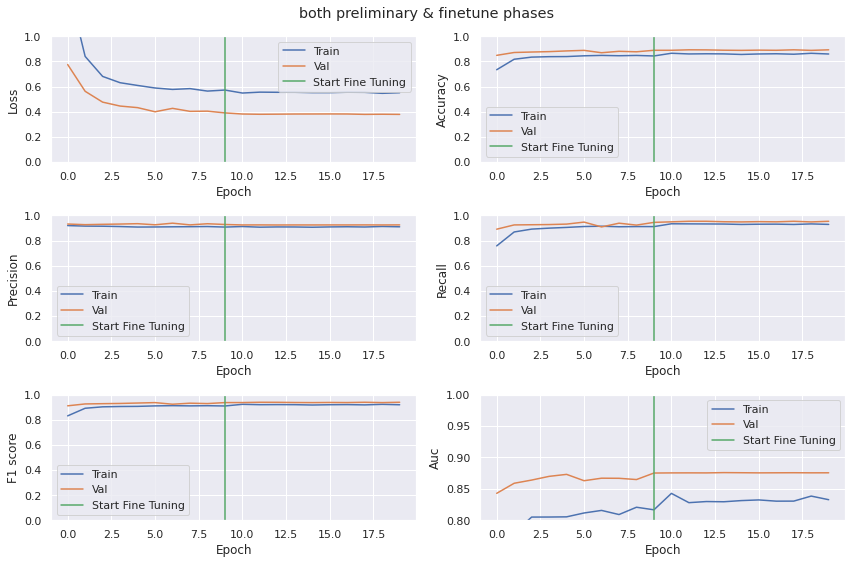
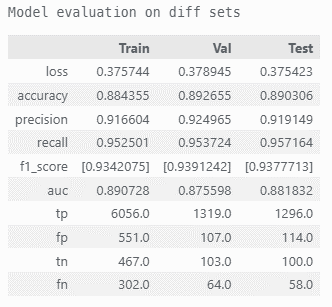
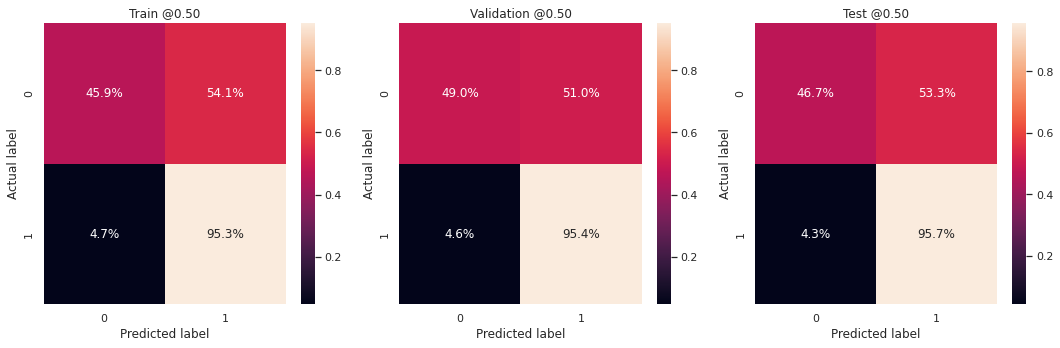


Figure 2: unfreeze some later layers (in base\_model) & re-train them







## MobileNetV2 (2022.08.10)

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 1736 |  |
| Train set | 1389 |  |
| Val set | 227 |  |
| Test set | 69 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | tekboart | base\_mode, tekboart |
| Dropout | 0.2 |  |
| L2 Regularization | None |  |
| Data Augmentation | YES (keras) |  |
| Class weight? | Yes |  |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=100, stair\_case=T |  |
| #epochs | 20 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | EarlyStopping(patience=5, ‘val\_loss’) |  |
| lr\_rate (for fine-tune) | 0.001 / 100 |  |
| lr\_schedule\_fine-tune | No |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | No |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Performance Measure | Train | Val | Test |
| Loss | .0914 | .3489 | .1720 |
| Accuracy | .9647 | .8728 | .9375 |

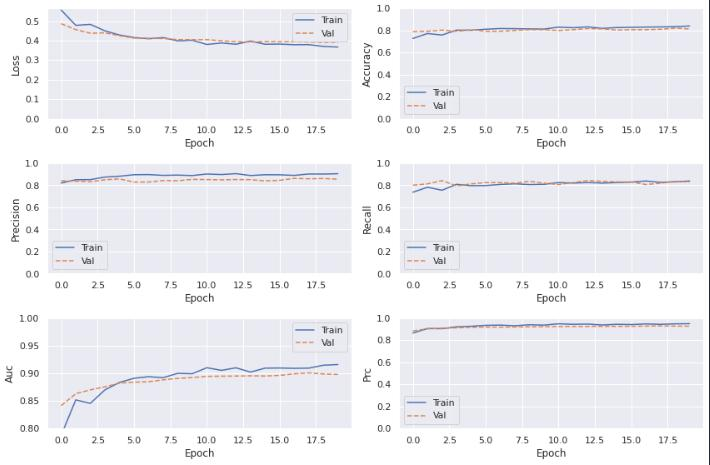


Figure 3: train the self added layers (i.e., Dense(1))

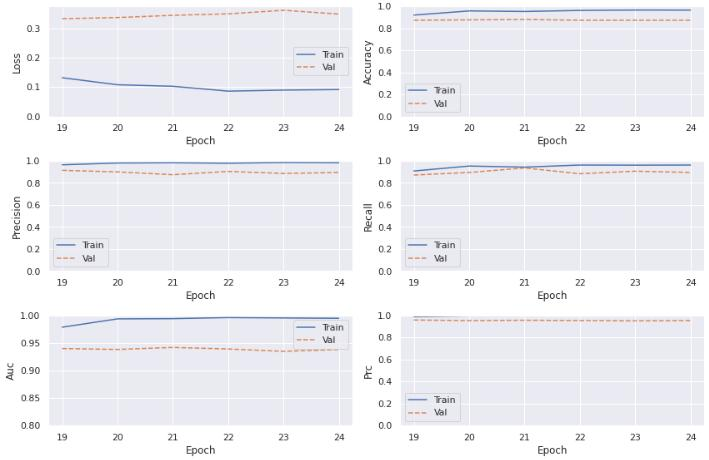
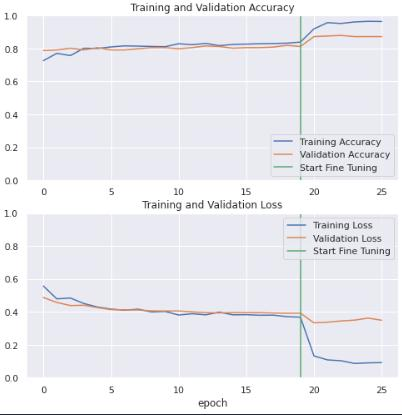
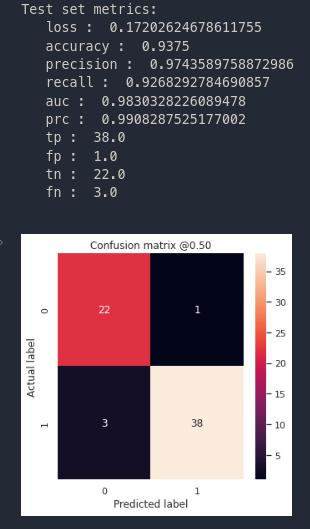


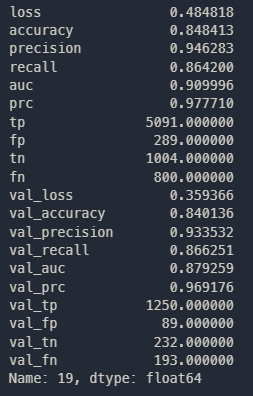
Figure 4: unfreeze some later layers (in base\_model) & re-train them

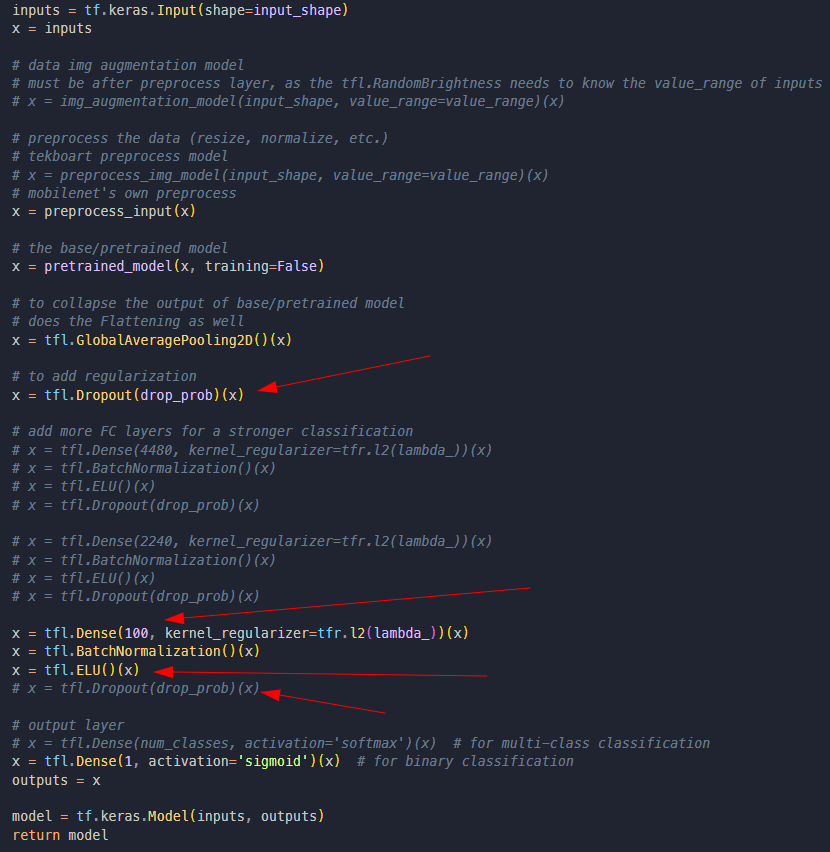




## MobileNetV2 (2022.08.25)

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 8980 |  |
| Train set | 7184 |  |
| Val set | 1796 |  |
| Test set | 0 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0 |  |
| Data Augmentation | NO (keras) |  |
| Class weight? | Yes | {0: 2.78, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 20 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| lr\_rate (for fine-tune) | 1e-8 |  |
| lr\_schedule\_fine-tune | No |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) |  |  |





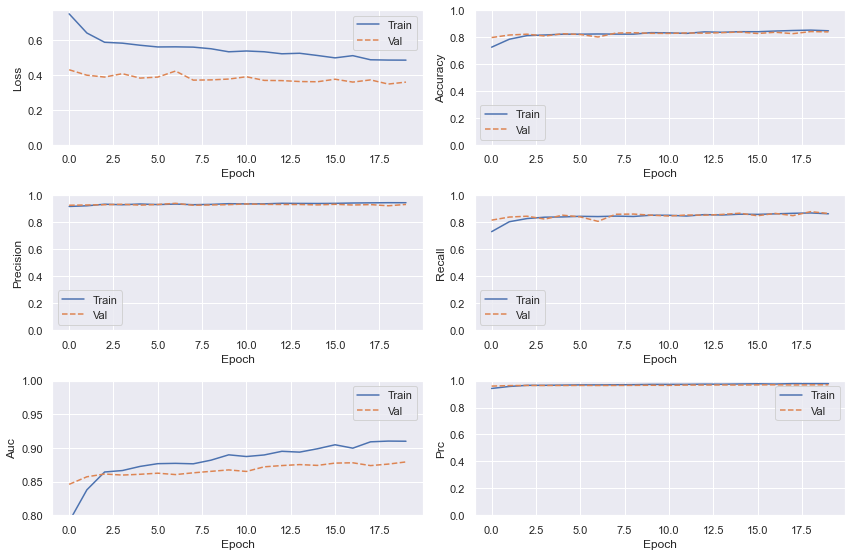


Figure 5: train the self added layers (i.e., Dense(1))

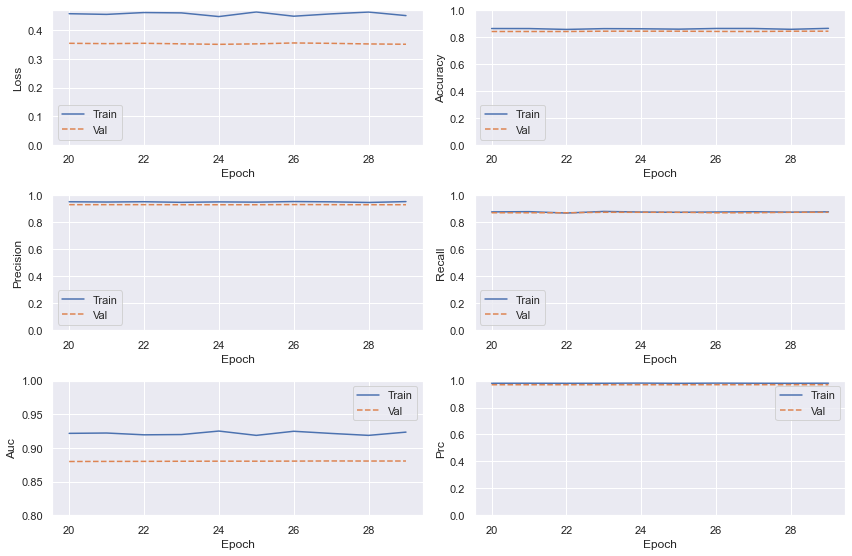
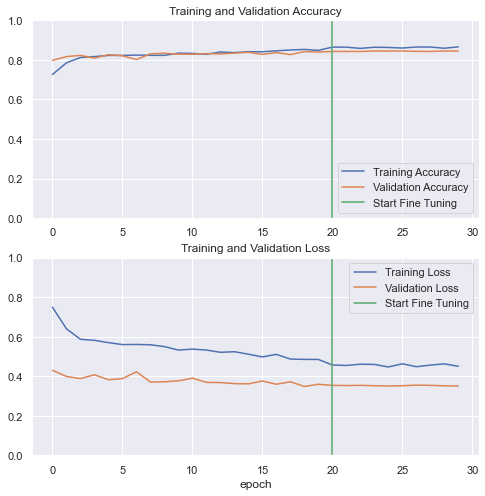
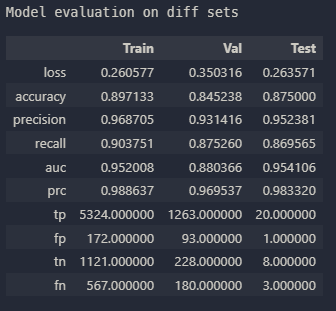
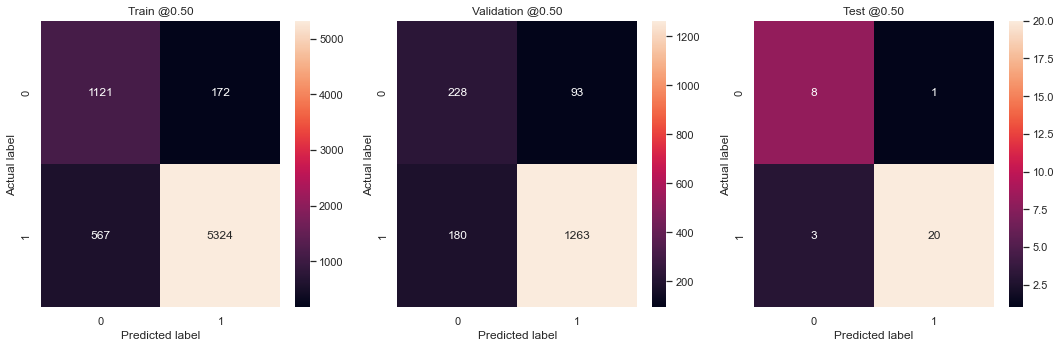


Figure 6: unfreeze some later layers (in base\_model) & re-train them

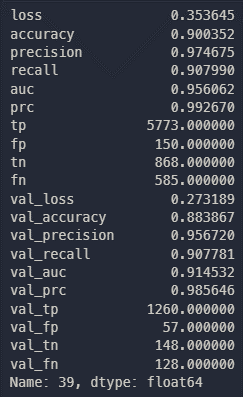


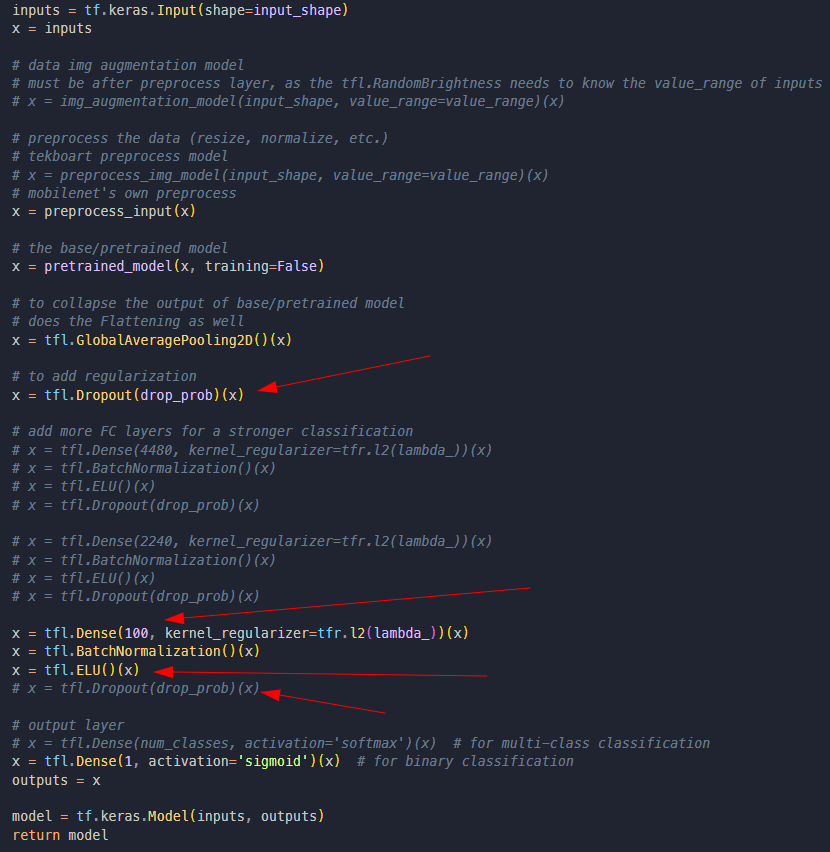




## MobileNetV2 (2022.08.26)

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0 |  |
| Data Augmentation | - |  |
| Class weight? | Yes | {0: 3.62, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 40 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? | Yes | {0: 3.62, 1: 1} |
| lr\_rate (for fine-tune) | 1e-6 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





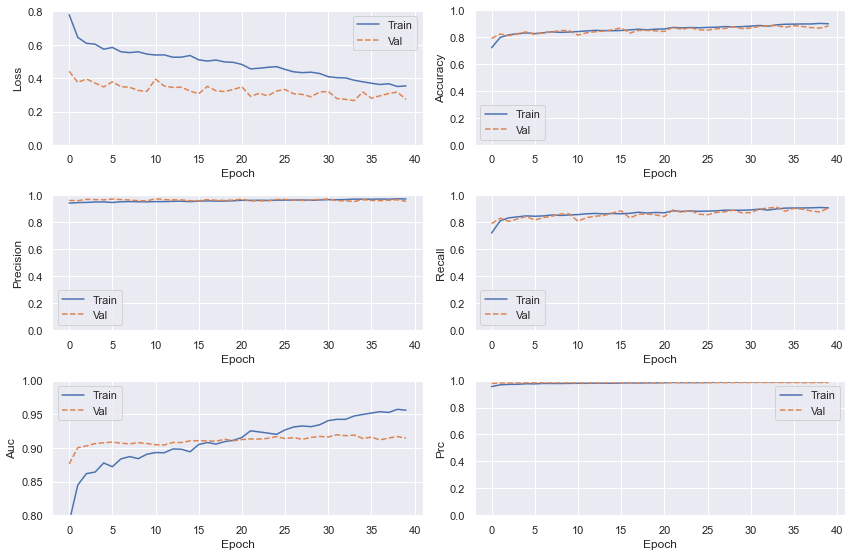


Figure 7: train the self added layers (i.e., Dense(1))

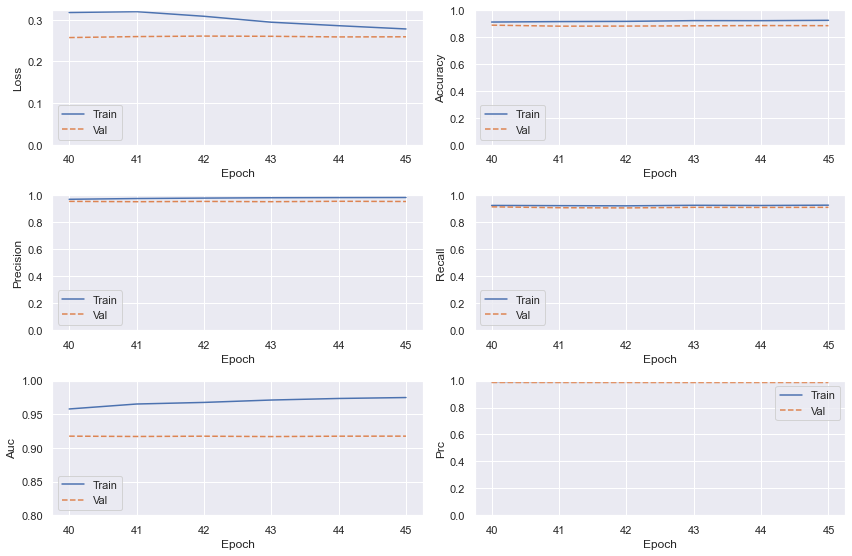
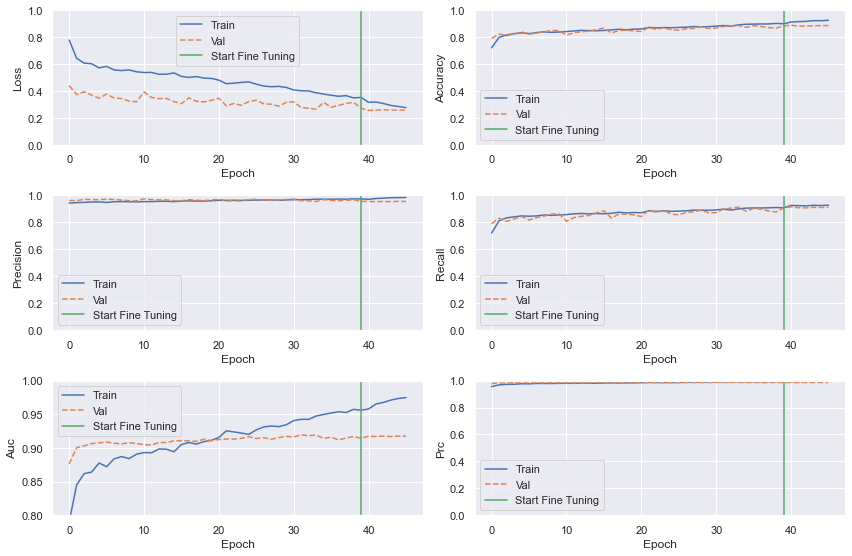
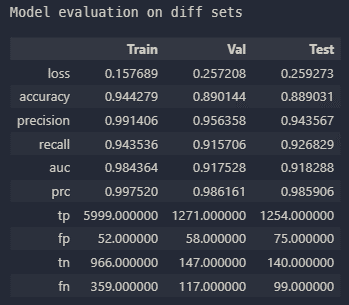
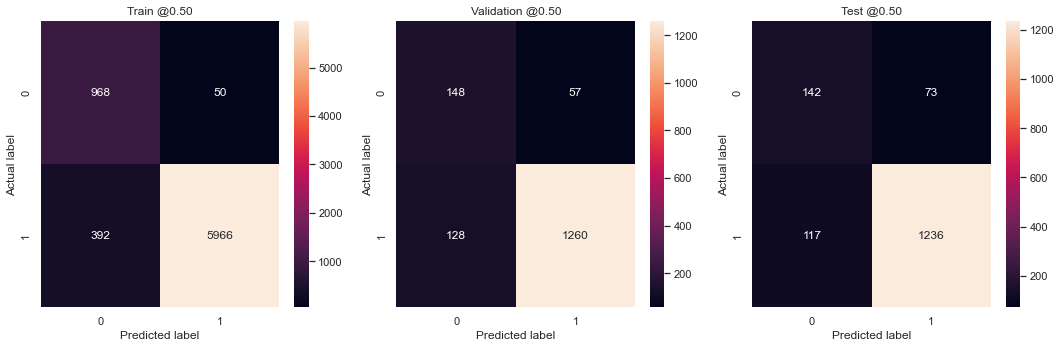


Figure 8: unfreeze some later layers (in base\_model) & re-train them

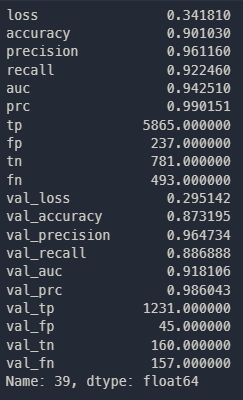






## MobileNetV2 (2022.08.26) 2

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2.62, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 40 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘auc’, ‘prc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2.62, 1: 1} |
| lr\_rate (for fine-tune) | 1e-6 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





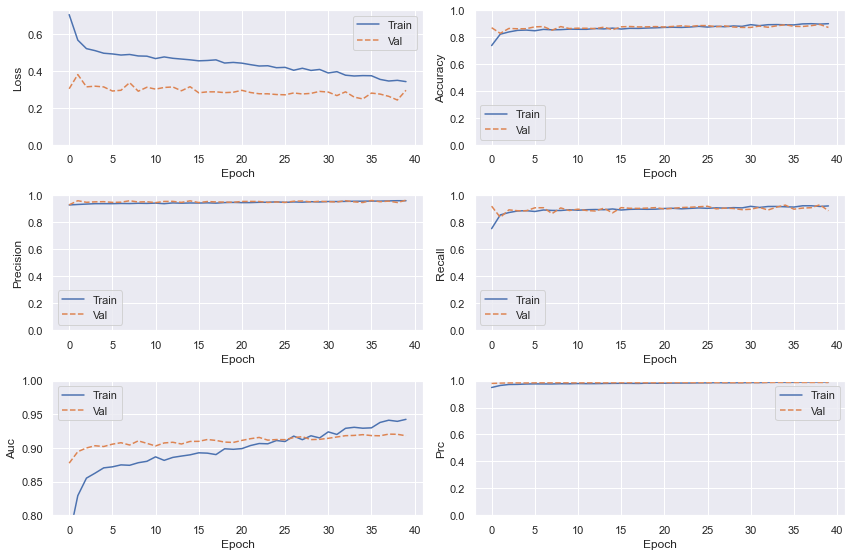


Figure 9: train the self added layers (i.e., Dense(1))

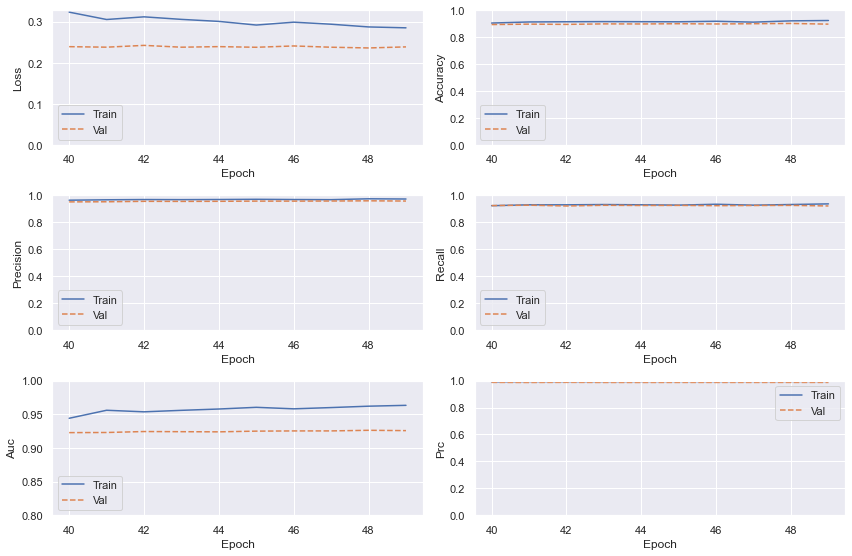
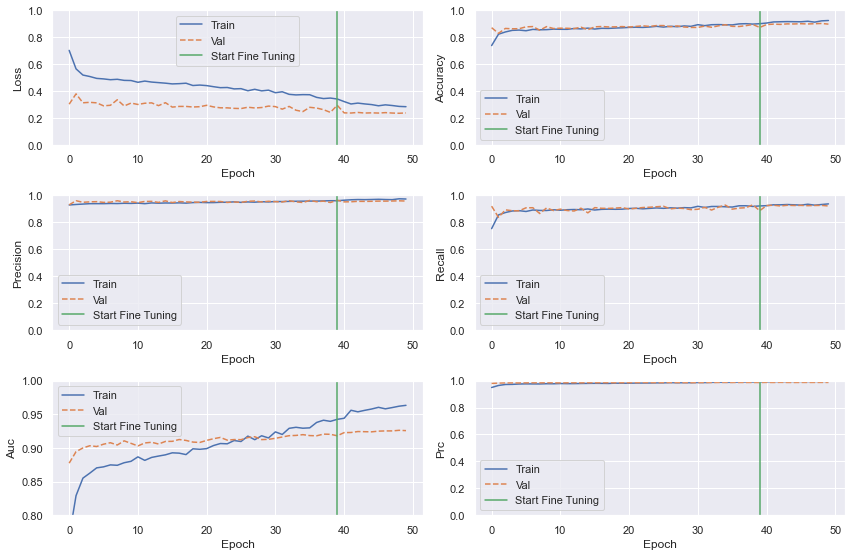
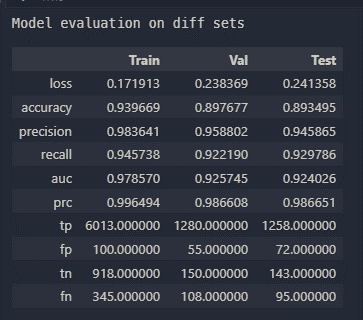
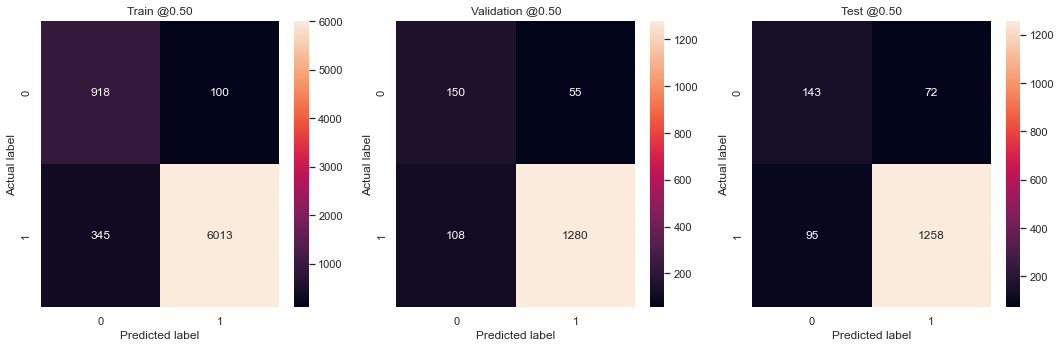


Figure 10: unfreeze some later layers (in base\_model) & re-train them

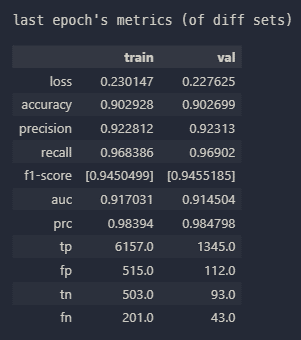






## MobileNetV2 (2022.08.27) (no class weight)

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | No | {0: 1, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 30 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | - |  |
| lr\_rate (for fine-tune) | - |  |
| lr\_schedule\_fine-tune | - |  |
| # un-freezed layers | - |  |
| #epochs (for fine-tune) | - |  |
| Callback (for fine-tune) | - |  |





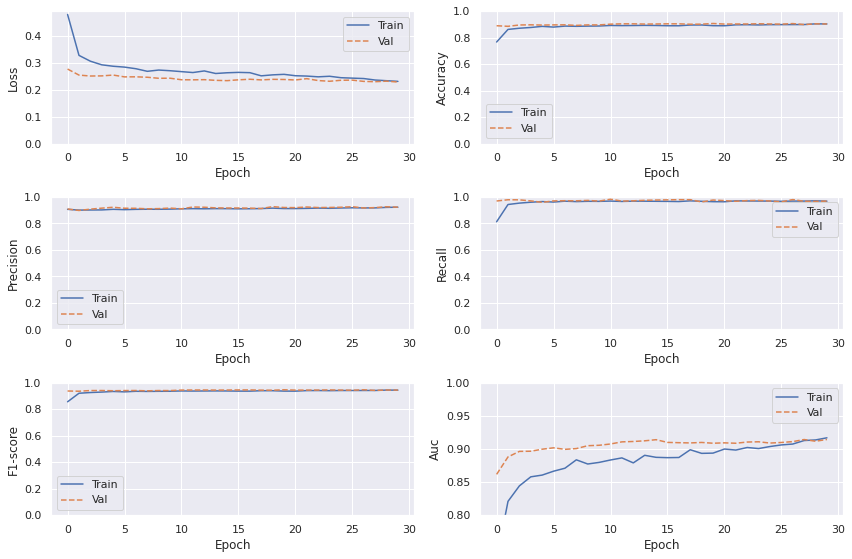


Figure 11: train the self added layers (i.e., Dense(1))

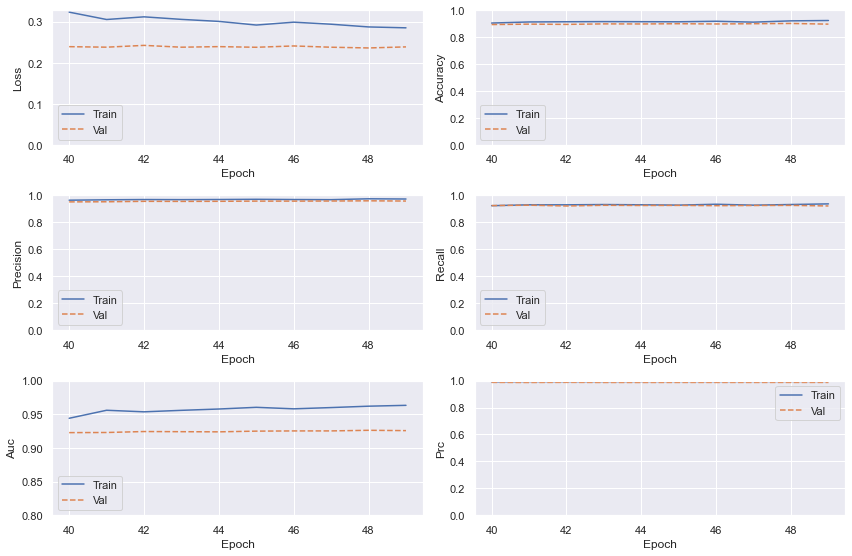
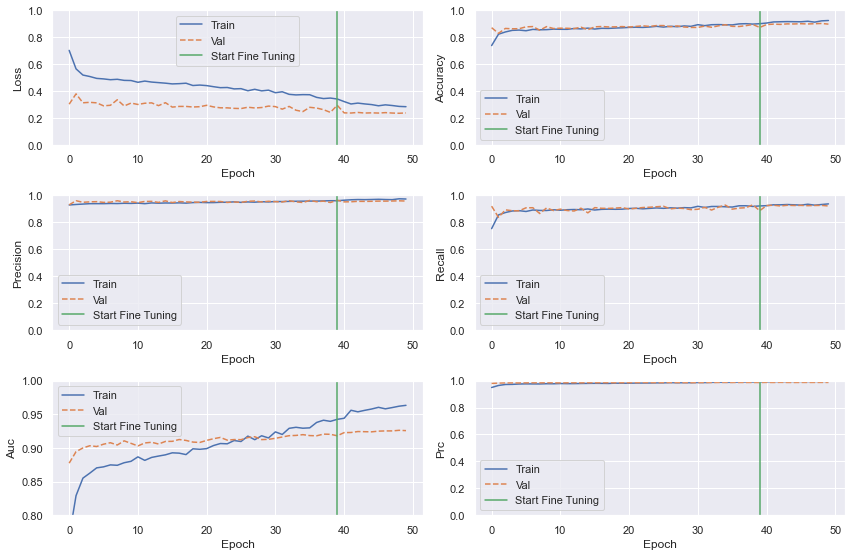
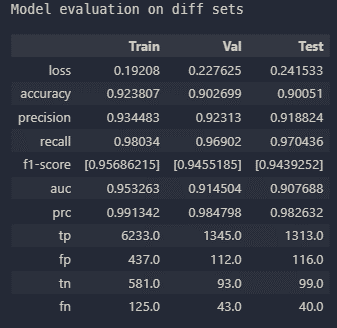
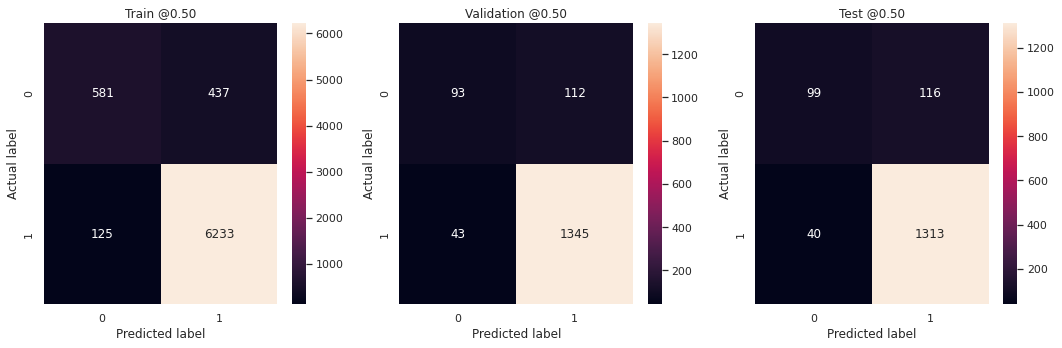


Figure 12: unfreeze some later layers (in base\_model) & re-train them

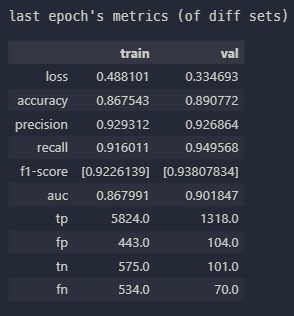






## MobileNetV2 (2022.08.31) ⭐

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 30 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2, 1: 1} |
| lr\_rate (for fine-tune) | 1e-6 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





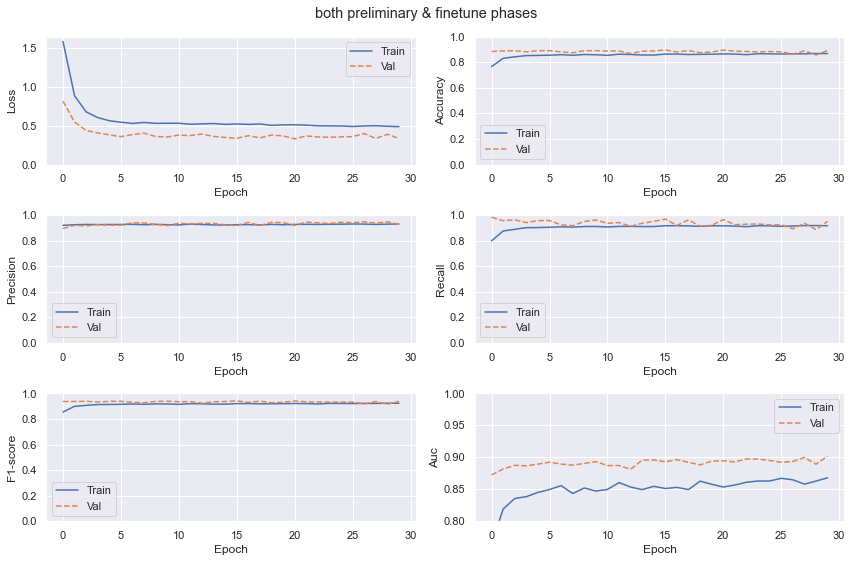


Figure 13: train the self added layers (i.e., Dense(1))

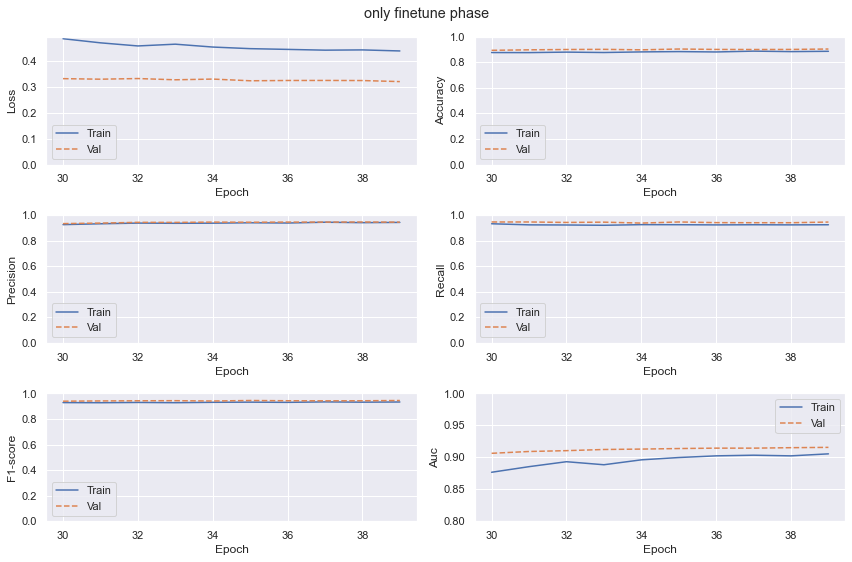
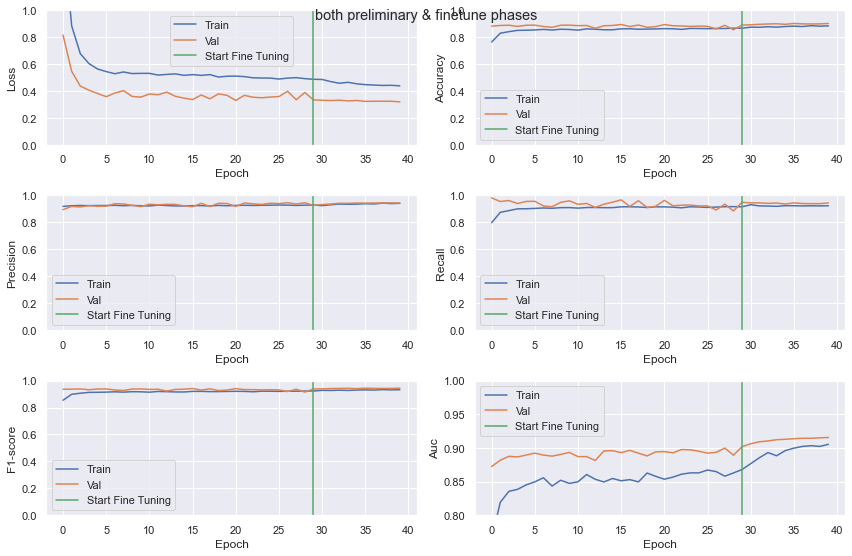
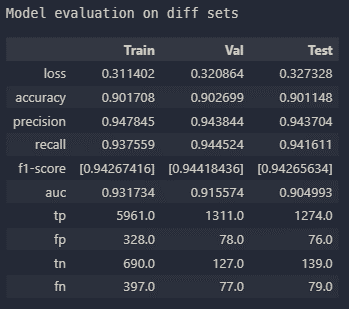
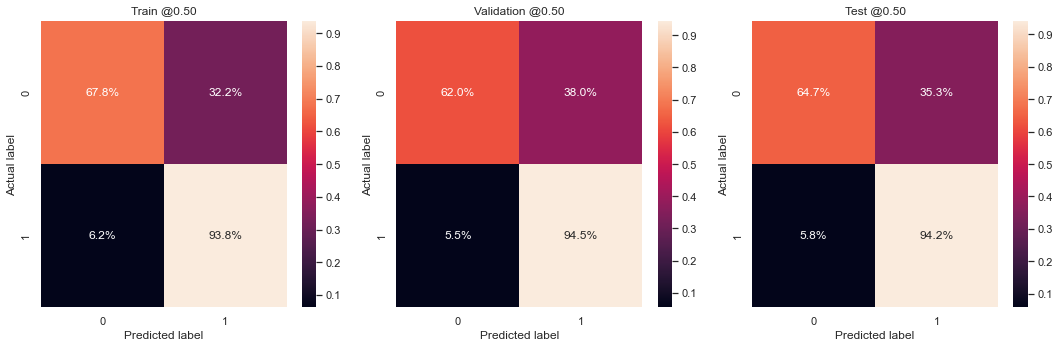


Figure 14: unfreeze some later layers (in base\_model) & re-train them

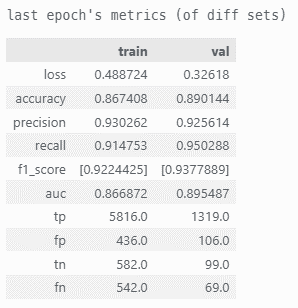






## MobileNetV2 (2022.09.05)

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 30 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2, 1: 1} |
| lr\_rate (for fine-tune) | 1e-6 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





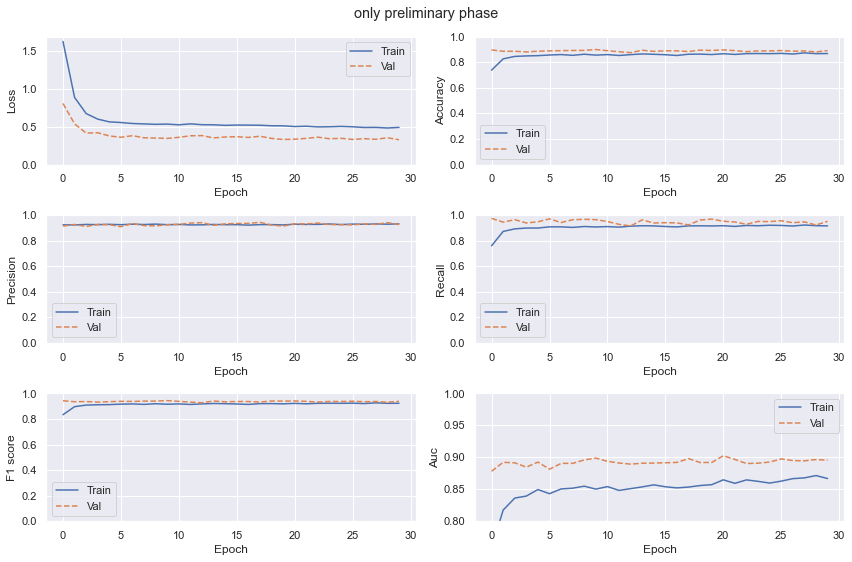


Figure 15: train the self added layers (i.e., Dense(1))

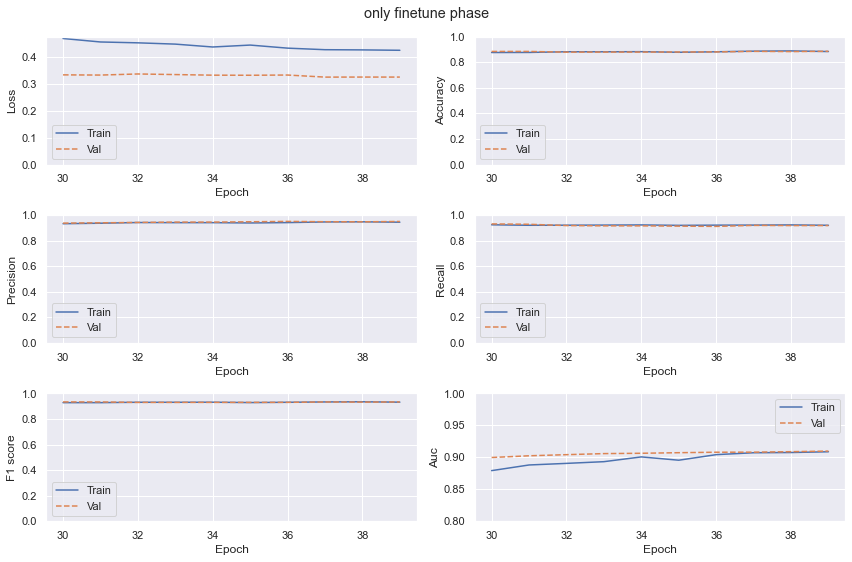
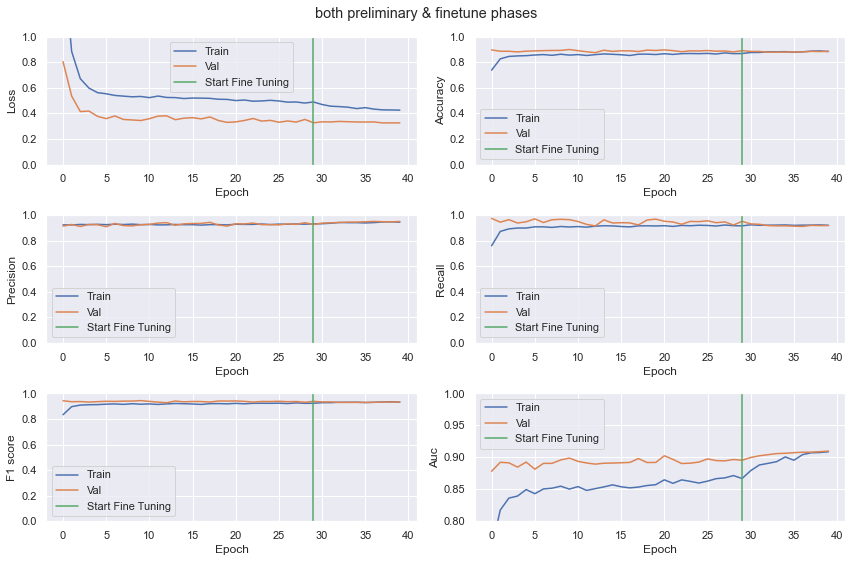
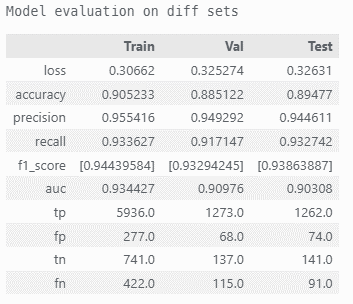
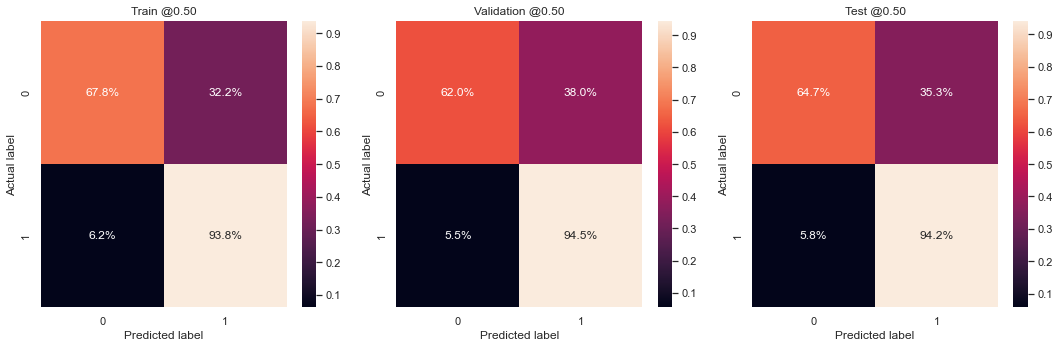


Figure 16: unfreeze some later layers (in base\_model) & re-train them

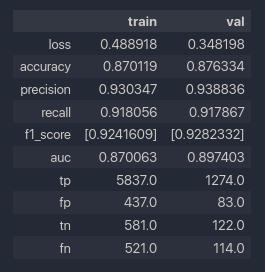






## MobileNetV2 (2022.10.25) ⭐

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.5 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | Yes (TF) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 30 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2, 1: 1} |
| lr\_rate (for fine-tune) | 1e-6 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +100 (54 out of 154) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





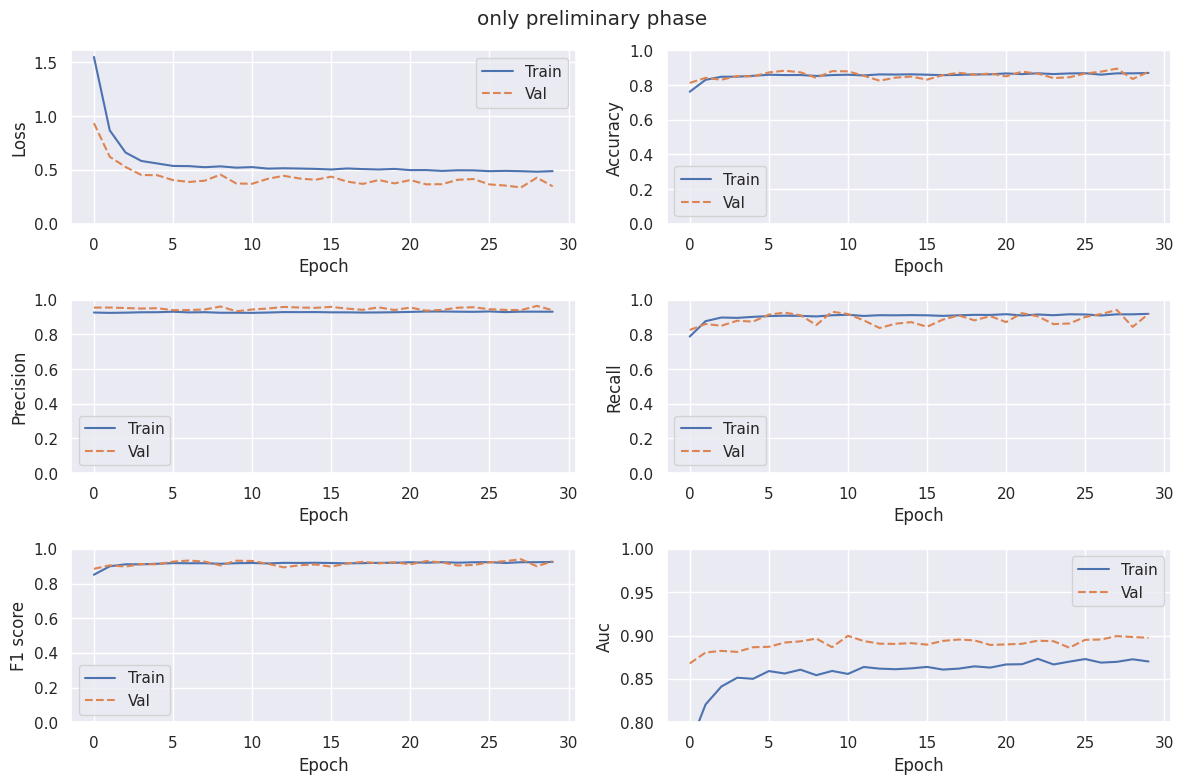


Figure 17: train the self added layers (i.e., Dense(1))

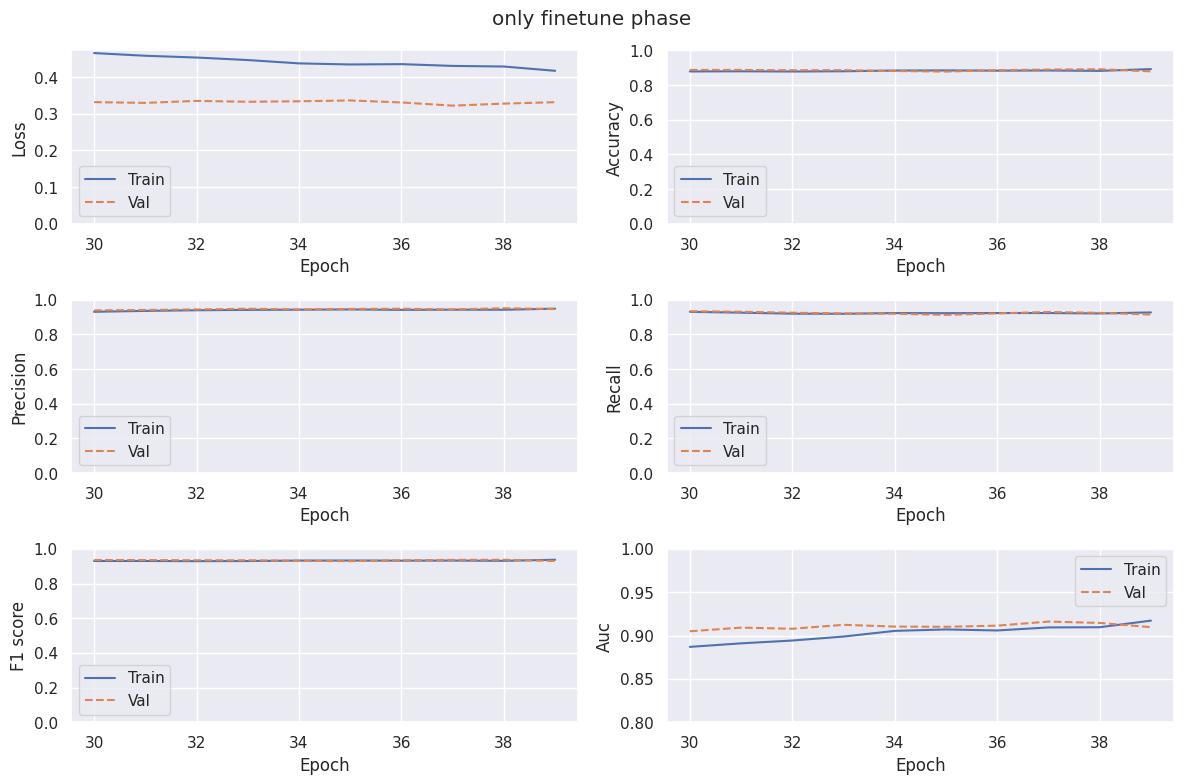
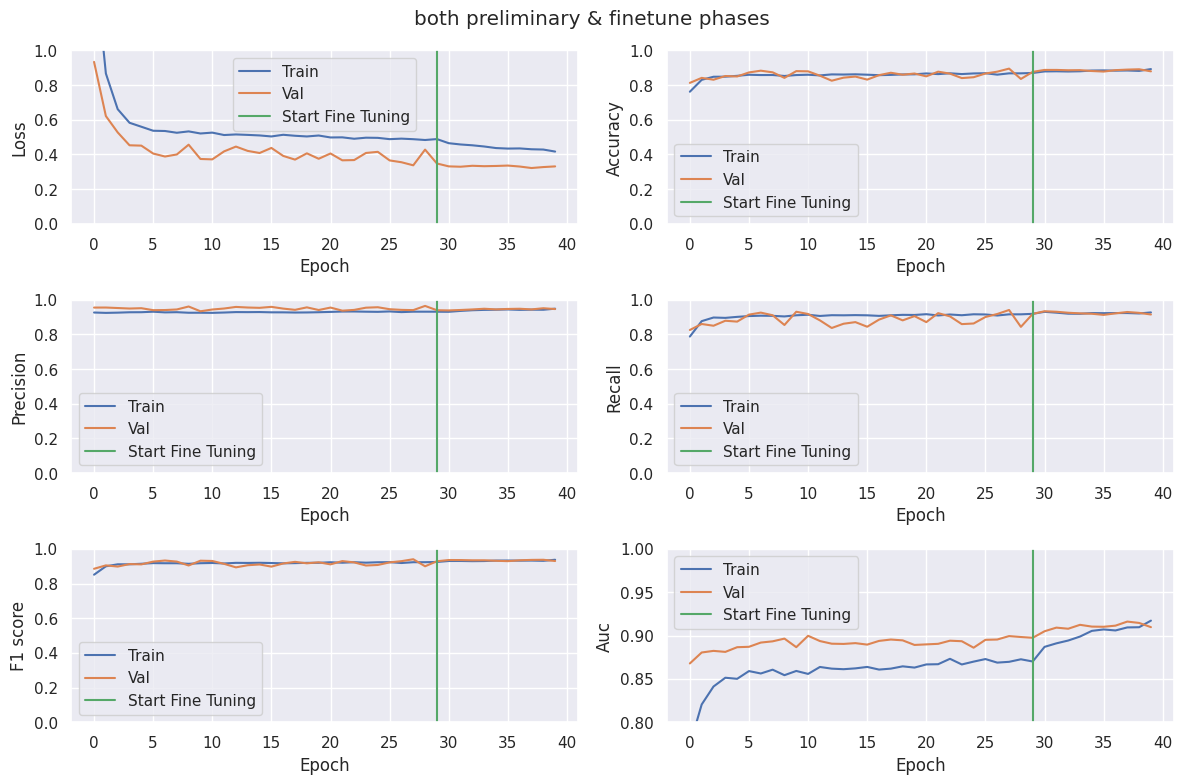
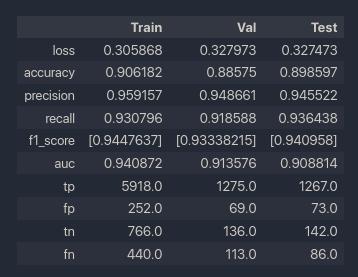
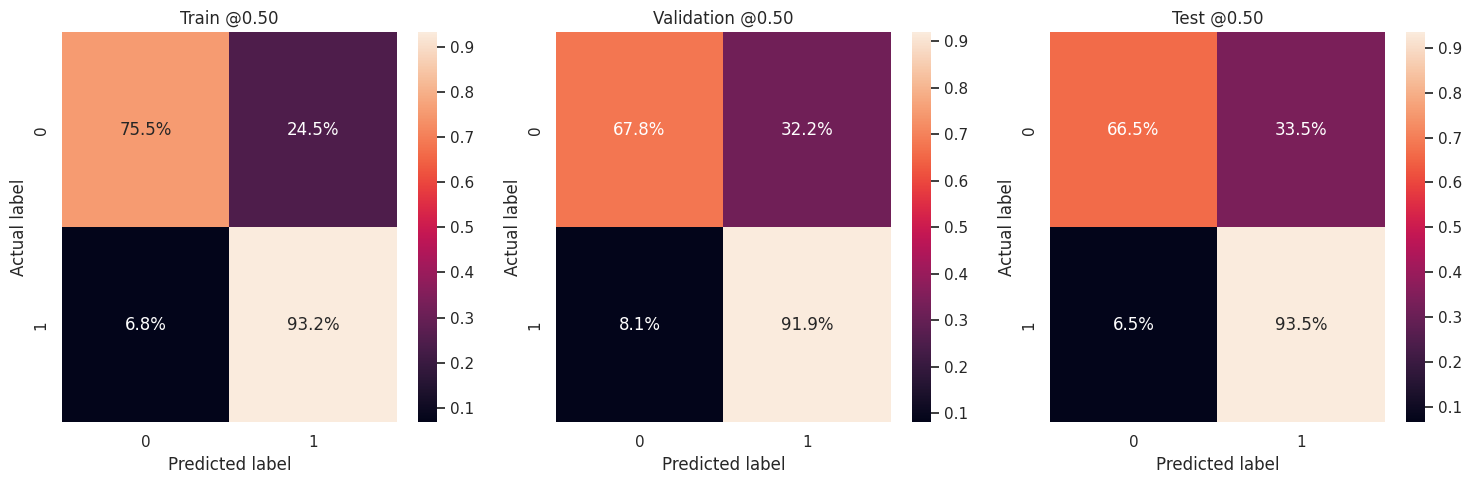


Figure 18: unfreeze some later layers (in base\_model) & re-train them

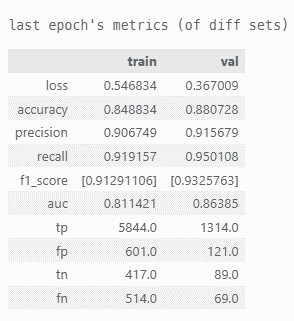






## InceptionResNet-v2 (2022.09.05) ⭐

|  |  |  |
| --- | --- | --- |
| Property | value | Description |
| Model | MobileNetV2\_tl | Transfer Learning |
| #classes/labels | {0: NO, 1:YES} | Binary classification |
| Total #images | 10537 |  |
| Train set | 7376 |  |
| Val set | 1593 |  |
| Test set | 1559 |  |
| batch\_size | 32 |  |
| input\_shape | (224, 224) | (64, 64), (150, 150), (224, 224) |
| normalization | [-1, 1] | [0, 1] / [-1, 1] |
| pre-process | MobileNet’s | base\_mode, tekboart |
| Dropout | 0.6 |  |
| L2 Regularization | 0.01 |  |
| Data Augmentation | Yes (keras) |  |
| Class weight? | Yes | {0: 2, 1: 1} |
| lr\_rate | 0.001 |  |
| lr\_schedule | decay\_step=1000, DR=.96, stair\_case=T |  |
| #epochs | 30 |  |
| Metrics | ‘loss’, ‘accuracy’, ‘precision’, ‘recall’, ‘f1-score’, ‘auc’, ‘tp’, ‘fp’, ‘tn’, ‘fn’ |  |
| callback | - |  |
| Class weight? (for fine-tune) | Yes | {0: 2, 1: 1} |
| lr\_rate (for fine-tune) | 1e-8 |  |
| lr\_schedule\_fine-tune | decay\_step=300, DR=.96, stair\_case=T |  |
| # un-freezed layers | +700 (80 out of 780) | Train layers from 100 and up |
| #epochs (for fine-tune) | 10 |  |
| Callback (for fine-tune) | - |  |





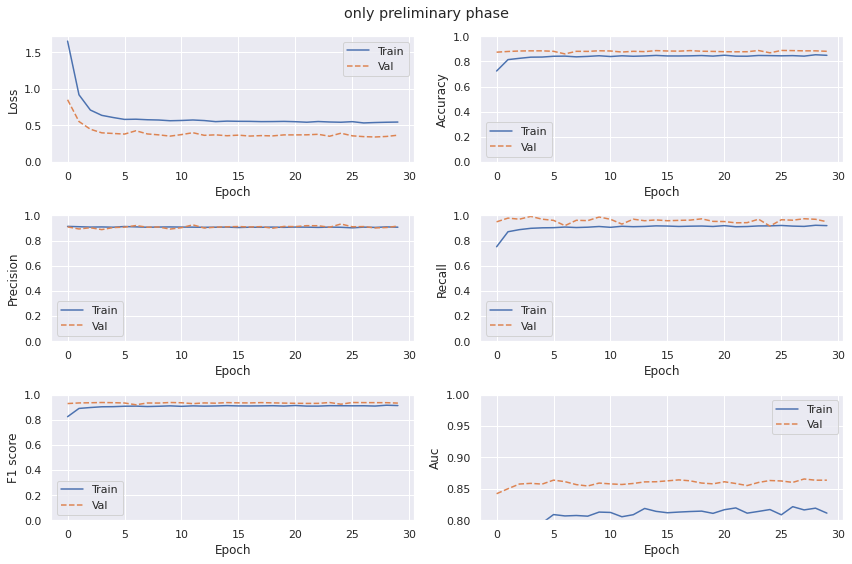


Figure 19: train the self added layers (i.e., Dense(1))

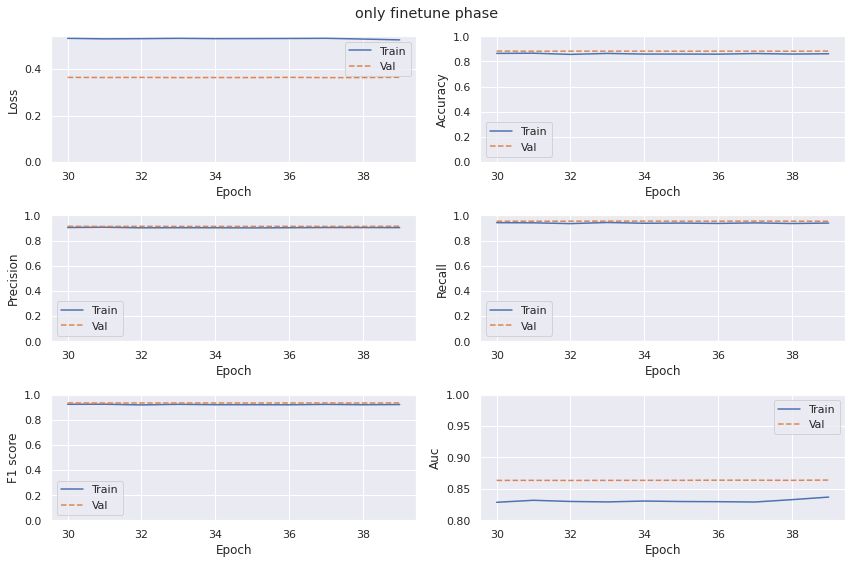


Figure 20: unfreeze some later layers (in base\_model) & re-train them

