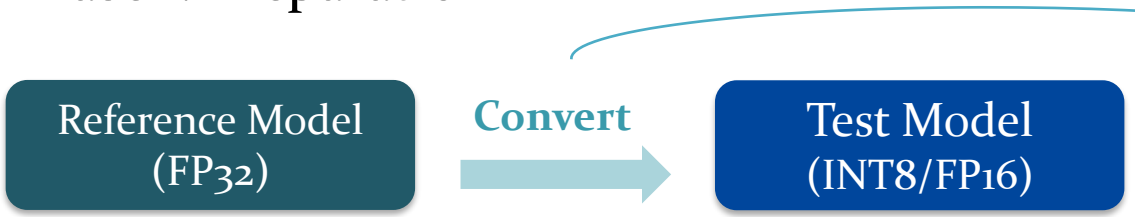
# Test Model Validation as A Precondition

For TS.53 Section 3 Hardware Performance: TOPS & TOPS/w Test

Sijing Cui

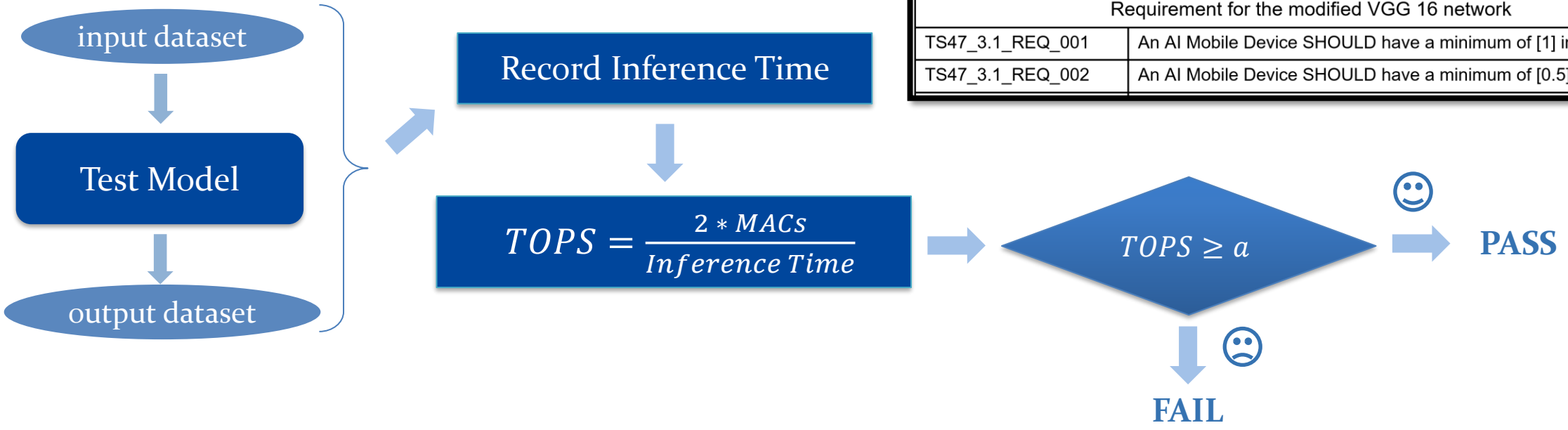
# Current TOPS Test Procedures in TS.53

## Phase I. Preparation




| | |
|---|---|
| TS47_3.2_REQ_004 | An AI Mobile Device SHOULD provide an SDK to convert DNN models from an existing format to the native format of the AI mobile device. Non-exhaustive examples of DNN model file format are: *.ckpt or *.pb, *.tflite, *.prototxt, *.pb or *.pth or *.pt, *.json and *.onnx. |

## Phase II. TOPS Test




| Requirement for the modified VGG 16 network | |
|---|---|
| TS47_3.1_REQ_001 | An AI Mobile Device SHOULD have a minimum of [1] int8 TOPS. |
| TS47_3.1_REQ_002 | An AI Mobile Device SHOULD have a minimum of [0.5] float16 TOPS. |

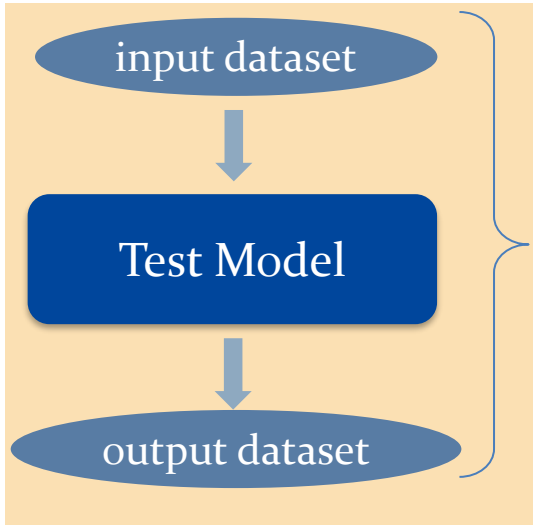
# Potential Issues in TOPS Measurement

## Phase I. Preparation


**Inconsistency**

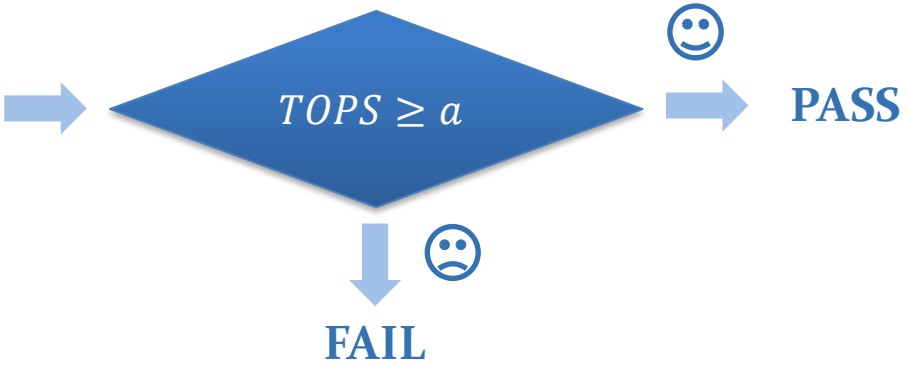
1. **Model.** vendors have specific SDK for model compression, causing models to have different formats, parameters or even structures. Leaving rooms for over-compression to speed up.

## Phase II. TOPS Test




Record Inference Time

$$TOPS = \frac{2 * MACs}{Inference\ Time}$$




2. **Execution.** DUTs' design vary (software and hardware), leading to different processing on computation. Making the computation process untransparent, some operations might be skipped during processing.

# A Validation as Precondition is in Demand

## Phase I. Preparation

Reference Model (FP32) → **Step 1. Convert** → Model to be check: Model_t (INT8/FP16) → **Step 2. Validate** → Test Model (INT8/FP16)

Take the model and the execution environment as a black box and check the output.

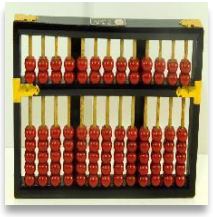
### Step 2. Validation

DUT environment

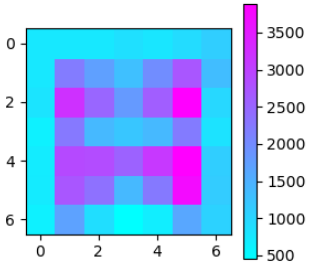
input dataset → Model_t (INT8/FP16) → output dataset

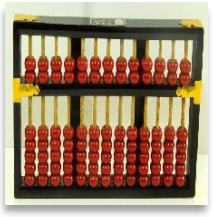
Utilize more information in the validation, thus focus on the output distribution.

VGG16_notop




Feature Tensor (7,7,512) → visualize / sum along z-axis → Activation Map (7,7)




Probability Vector (1, 1000) → decode → Label 'abacus'

VGG16

# Precondition Validation Workflow

## Phase I. Preparation

Reference Model (FP32)

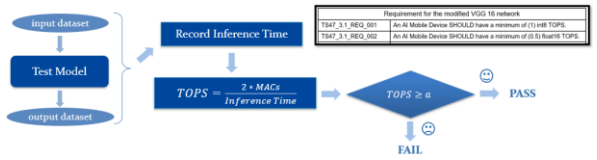
**Step 1. Convert**

Model_t (INT8/FP16)

**Step 2. Validate**

Test Model (INT8/FP16)

## Phase II. TOPS Test




## Step 2. Validation

input dataset

Reference Model (FP32)

Model_t (INT8/FP16)

reference output set

test output set

Compare Dataset

Validate

DUT

Trustworthy Device (E.g., PC)

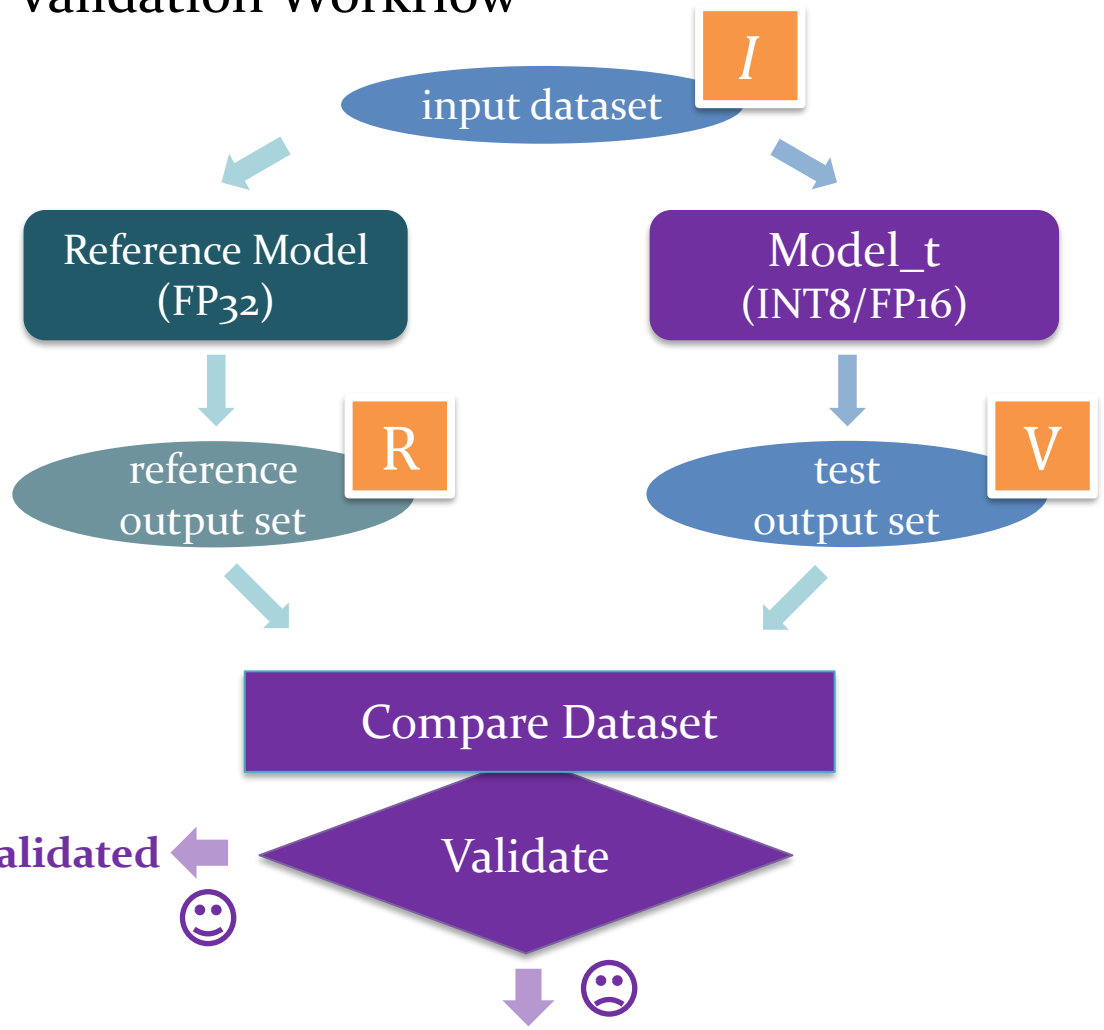
Take the FP32 reference model's output as baseline.

Check if the test outputs are too obscure (i.e. lose too much information) compared with the reference outputs

Further check on the converted model, the execution environment (scripts, SDK, etc.)

# Validation Procedures (1)

## Validation Workflow




| Step | Test procedure | Expected result |
|---|---|---|
| 1 | Number the data in Test Dataset from 1 to 1000. | Each test data is denoted as I(n), where n ∈ [1, 1000]. |
| 2 | Run Reference Model with Test Dataset on a PC. | The output dataset of PC is obtained. Each data inside is in tensor form, denoted as R(n) according to its input I(n). |
| 3 | Run Model_t with Test Dataset on DUT. | The output dataset of DUT is obtained. |
| 4 | Convert each DUT output data into a tensor, with the shape identical to the shape of R(n). | The converted DUT output data is in tensor form, denoted as V(n) according to its input I(n) in step 3. |

## Generate Output Set

- Input dataset

$$I = \{I_1, I_2, \dots, I_N\}$$

- Reference output set

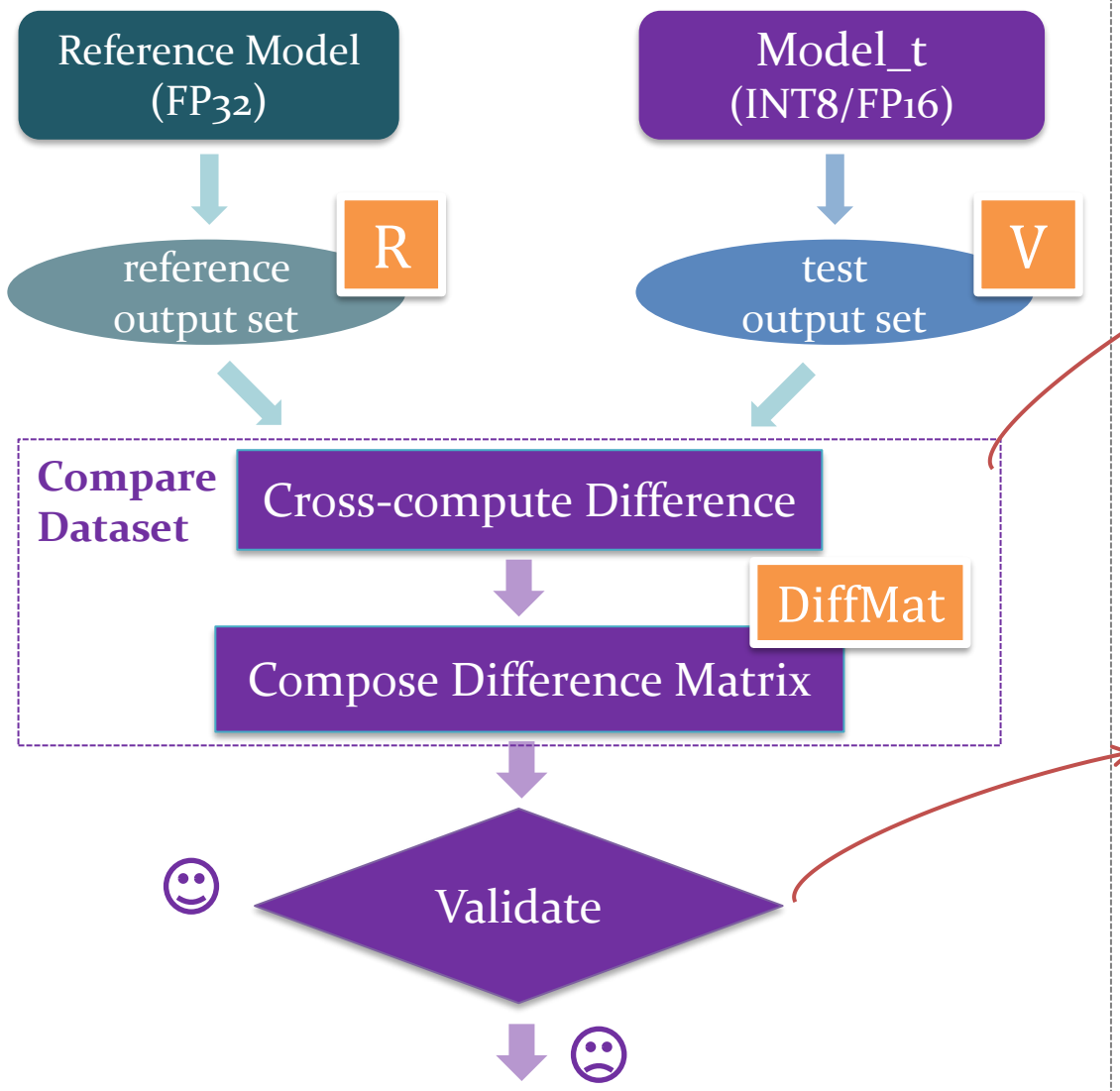
$$R = \{R_1, R_2, \dots, R_N \mid R_n = Reference\_Model(I_n),\ 1 \le n \le N\}$$

- Test output set

$$V = \{V_1, V_2, \dots, V_N \mid V_n = Model\_t(I_n), 1 \le n \le N\}$$

# Validation Procedures (2)

## Validation Workflow



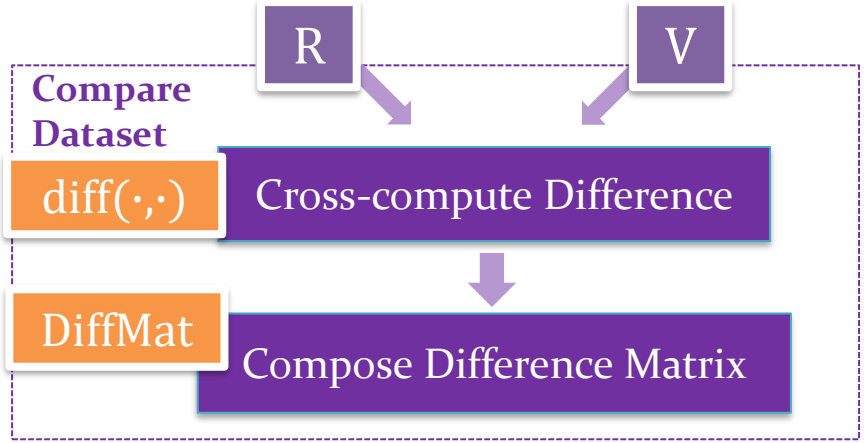

| Step | Test procedure | Expected result |
|---|---|---|
| 5 | Calculate the difference between a given V(n) and each R(m), m ∈ [1, 1000].<br>Note: Euclidean distance is recommended as the difference function. | A difference value set of the given V(n) is obtained. |
| 6 | Repeat step 5 until each V(n) is calculated. | A two-dimensional difference matrix is obtained, denoted as DiffMat. Each element DiffMat[m, n] represents the difference value between R(m) and V(n), where n, m ∈ [1, 1000]. |

| Step | Test procedure | Expected result |
|---|---|---|
| 7 | Count the diagonal elements that are the minimum of their own row, and calculate the proportion of minimum diagonal elements in all DiffMat[n, n]. | The proportion of minimum diagonal elements in all DiffMat[n, n] should be greater than [99%]. |
| 8 | Classify the DiffMat elements into two classes based on their difference values. Label the elements with strong similarity as Positive and the others as Negative. | All elements in DiffMat are sorted and labelled. |
| 9 | Take the Positive diagonal elements as True Positive instances, and calculate the F1-score of the classification. Denote the result as Fc%. (The information and utility preserved in Model_t's output are similar to those preserved in Reference Model's output, under the confidence of Fc%.) | If Fc% is not lower than [95%], it is considered that Model_t and Reference Model have a strong similarity, and Model_t can be validated as Test Model. |

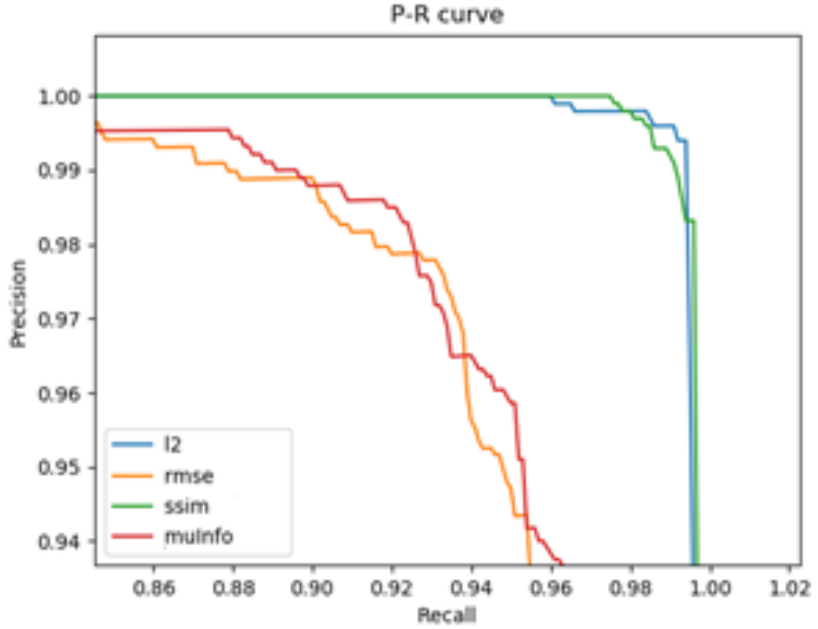
# Metrics to Calculate Output Difference

Reference output set $R = \{R_1, R_2, \dots, R_N \mid R_n = Reference\_Model(I_n),\ 1 \le n \le N\}$

Test output set $V = \{V_1, V_2, \dots, V_N \mid V_n = Model\_t(I_n), 1 \le n \le N\}$




- Which difference metric could distinguish the diagonals?




| Candidate Metric | Object | Object Shape |
|---|---|---|
| L2-distance (Euclidean distance) | feature tensor | (7, 7, 512) |
| muInfo (mutual Information) | | |
| SSIM (Structural Similarity) | activation map | (7, 7) |
| RMSE (Root Mean Squard Error) | | |

- L2 distance is a recommended difference metric $diff(\cdot,\cdot)$

Feature Tensor (7,7,512) → sum along z-axis → Activation Map (7,7)

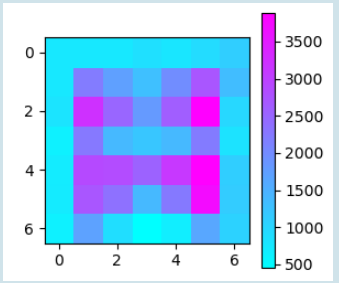
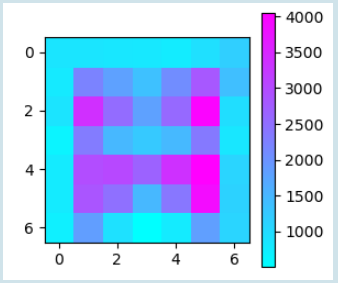
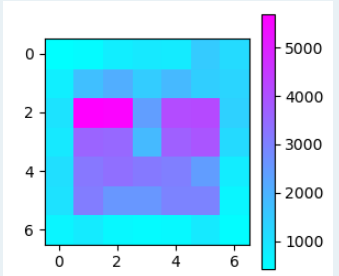
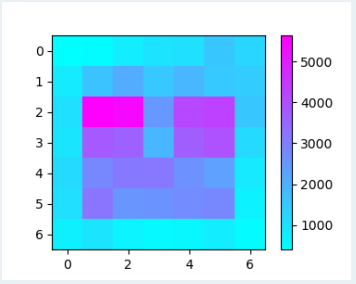
$$d(x,y) := \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}.$$

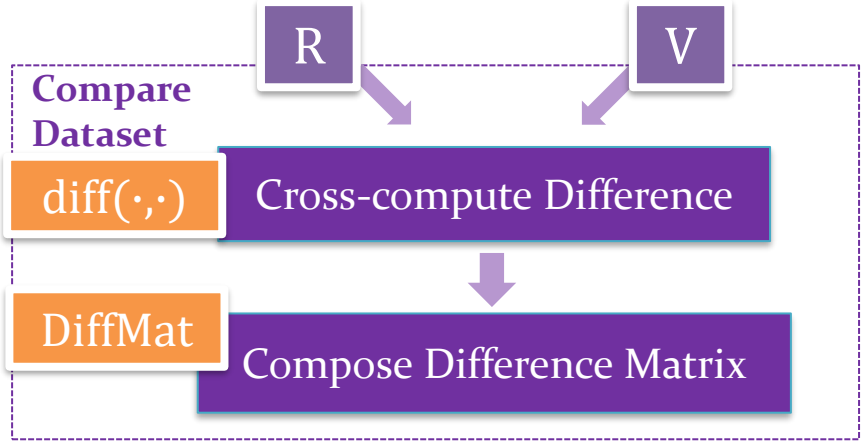
In VGG_notop test case, $n = 7 \times 7 \times 512$

# Cross-Comparison on Model Output Set

Reference output set $R = \{R_1, R_2, \dots, R_N \mid R_n = Reference\_Model(I_n),\ 1 \le n \le N\}$

Test output set $V = \{V_1, V_2, \dots, V_N \mid V_n = Model\_t(I_n), 1 \le n \le N\}$

| Input Image | Reference Output | Test Output |
|---|---|---|
|  $I_1$ |  $R_1$ |  $V_1$ |
|  $I_2$ |  $R_2$ |  $V_2$ |

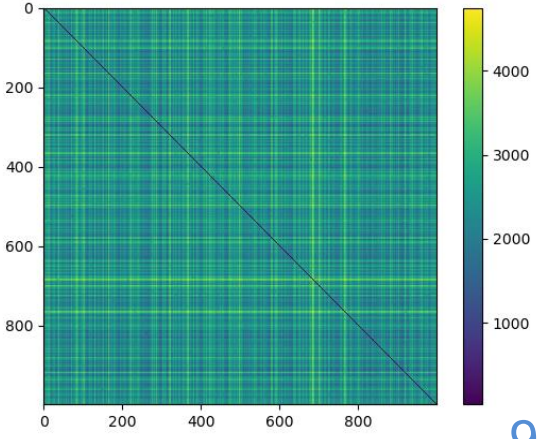



$$DiffMat[m,n] = diff(R_m, V_n), 1 \le m, n \le N.$$

difference funcion/metric $-$ $diff(\cdot,\cdot)$

| | V1 | V2 | ... | VN |
|---|---|---|---|---|
| R1 | diff(R1, V1) | diff(R1, V2) | ... | diff(R1, VN) |
| R2 | diff(R2, V1) | diff(R2, V2) | ... | diff(R2, VN) |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| RN | diff(RN, V1) | diff(RN, V2) | ... | diff(RN, VN) |

DiffMat



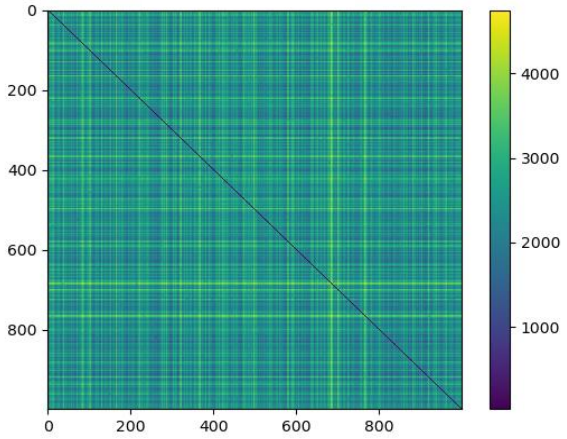
- The outputs with identical index share similar distribution.

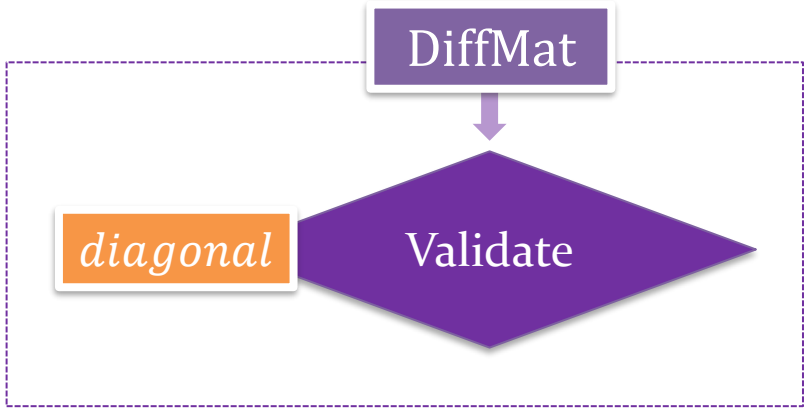
**Strong Similarity → Strong Diagonals in DiffMat**

# Examine the Diagonals in L2-DiffMat (1)

- If everything runs perfectly/normally, $V_n$ will have $R_n$ as its nearest reference, making the l2-distance between them the minimal.

| | V1 | V2 | ... | VN |
|---|---|---|---|---|
| R1 | diff(R1, V1) | diff(R1, V2) | ... | diff(R1, VN) |
| R2 | diff(R2, V1) | diff(R2, V2) | ... | diff(R2, VN) |
| ⋮ | ⋮ | ⋮ | ⋱ | ⋮ |
| RN | diff(RN, V1) | diff(RN, V2) | ... | diff(RN, VN) |

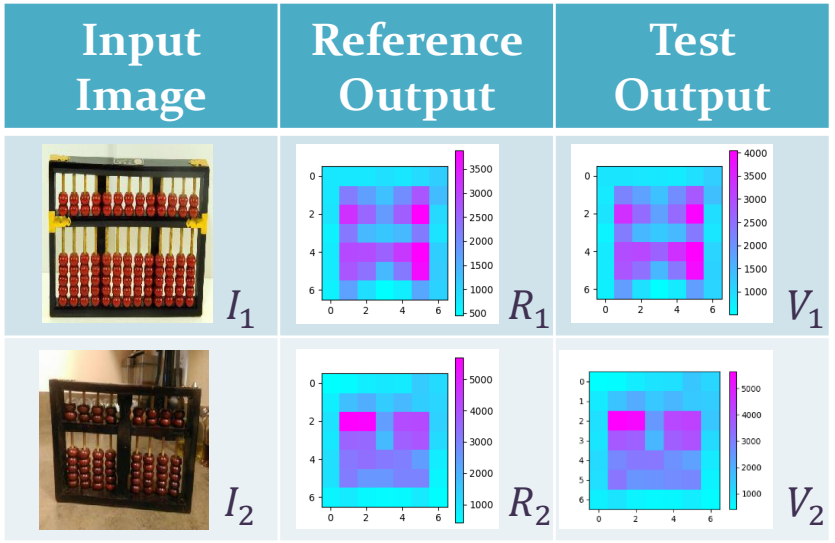



- Examination 1: Check if $R_n$ is the closest reference of $V_n$.

  → the Diagonals are the minimal of their own row.

| Step | Test procedure | Expected result |
|---|---|---|
| 7 | Count the diagonal elements that are the minimum of their own row, and calculate the proportion of minimum diagonal elements in all DiffMat[n, n]. | The proportion of minimum diagonal elements in all DiffMat[n, n] should be greater than [99%]. |




Reference output set

$$R = \{R_1, R_2, \dots, R_N \mid R_n = Reference\_Model(I_n),\ 1 \le n \le N\}$$

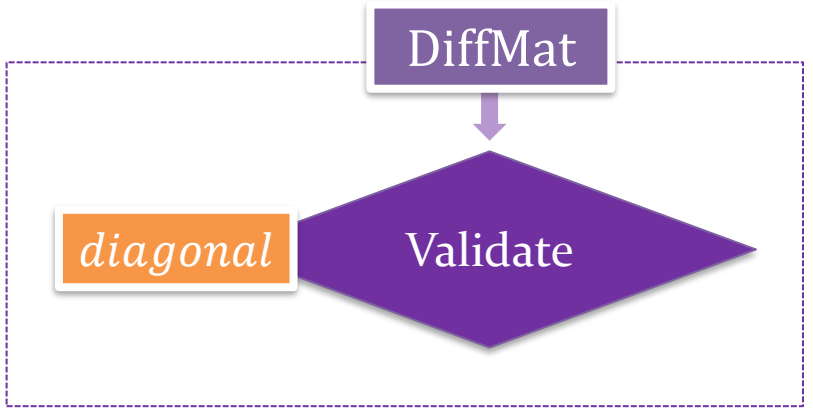
Test output set

$$V = \{V_1, V_2, \dots, V_N \mid V_n = Model\_t(I_n), 1 \le n \le N\}$$

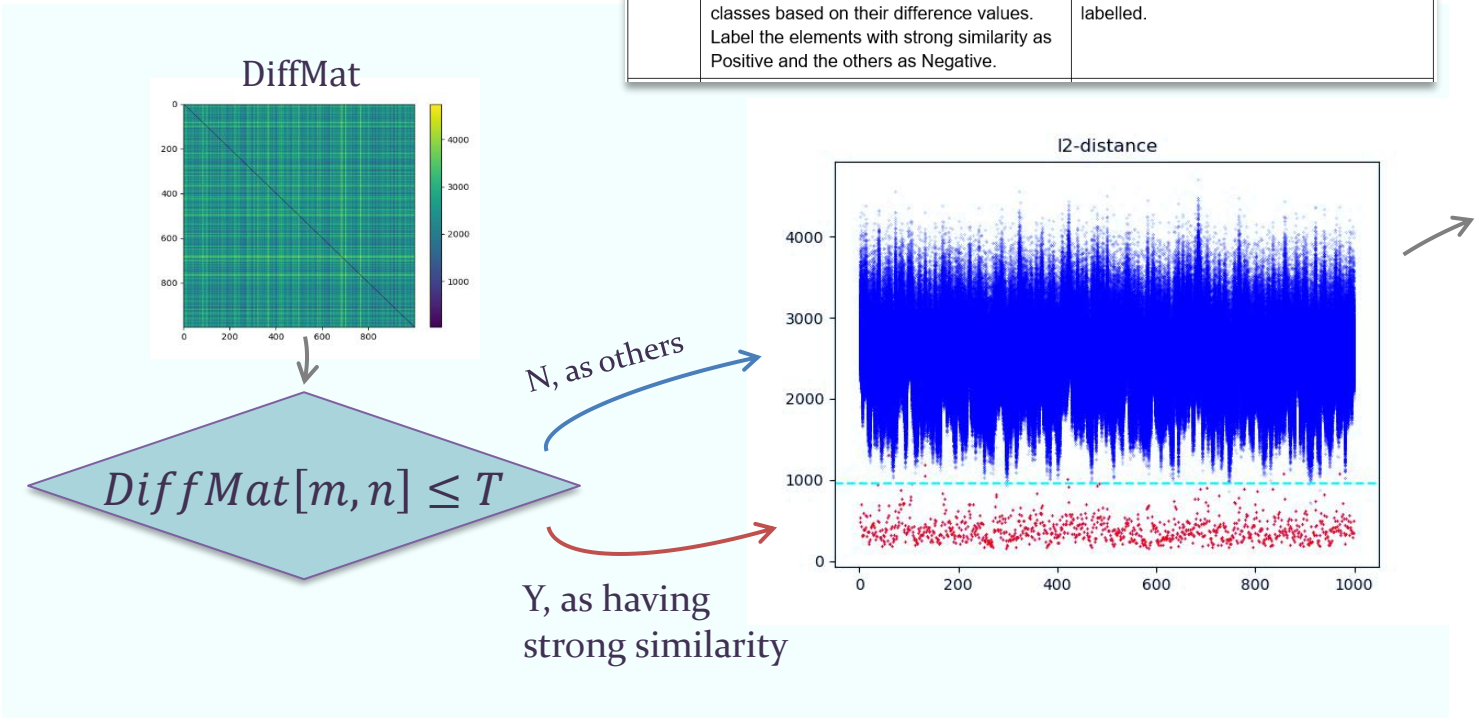
# Examine the Diagonals in L2-DiffMat (2)

- Examination 1: Check if $R_n$ is the closest reference of $V_n$.
- Examination 2: Check if $V_n$ is significantly closer to its reference $R_n$.

  → the Diagonal values stand out from all DiffMat elements because of their strong similarity.




| Step | Test procedure | Expected result |
|---|---|---|
| 8 | Classify the DiffMat elements into two classes based on their difference values. Label the elements with strong similarity as Positive and the others as Negative. | All elements in DiffMat are sorted and labelled. |




Evaluate the classification result

| | Diagonal | Non-Diagonal |
|---|---|---|
| Classified as having strong similarity | True Positive (TP) | False Negative (FN) |
| Classified as others | False Positive (FP) | True Negative (TN) |

$$Precsion\ P = \frac{TP}{TP + FP}$$

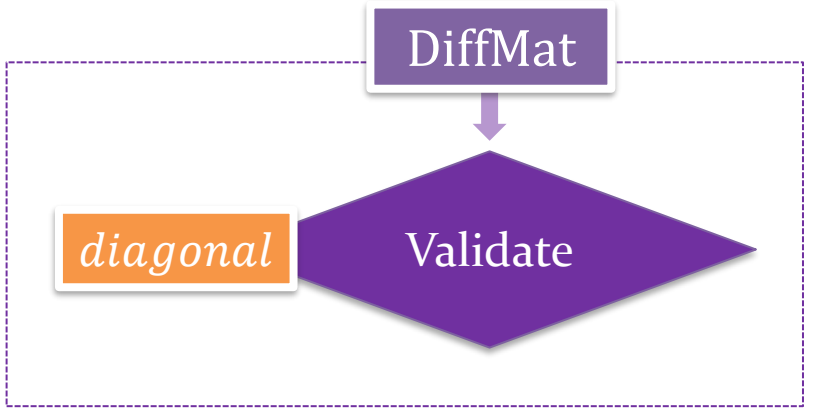
$$Recall\ R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2PR}{P + R}$$

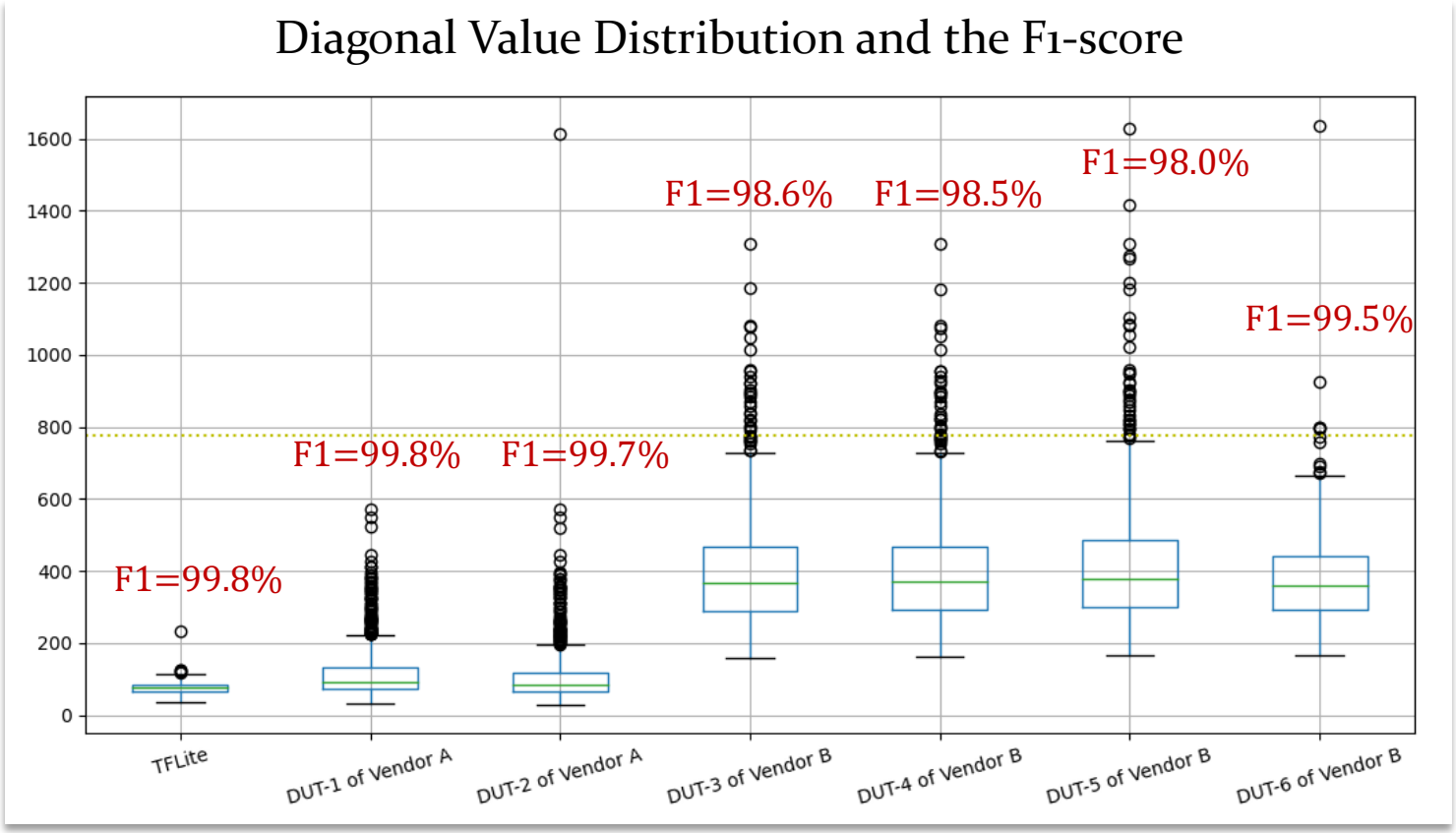
# Examine the Diagonals in L2-DiffMat (3)

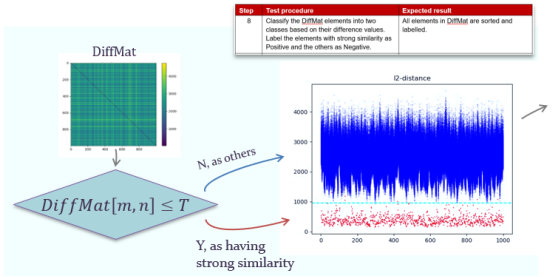
- Examination 2: Check if $V_n$ is significantly closer to its reference $R_n$.

  → the Diagonal values stand out from all DiffMat elements because of their strong similarity.




How well can the diagonals stand out ? - F1 Score




| | Diagonal | Non-Diagonal |
|---|---|---|
| Classified as having strong similarity | True Positive (TP) | False Negative (FN) |
| Classified as others | False Positive (FP) | True Negative (TN) |

$Precsion\ P = \frac{TP}{TP+FP}$

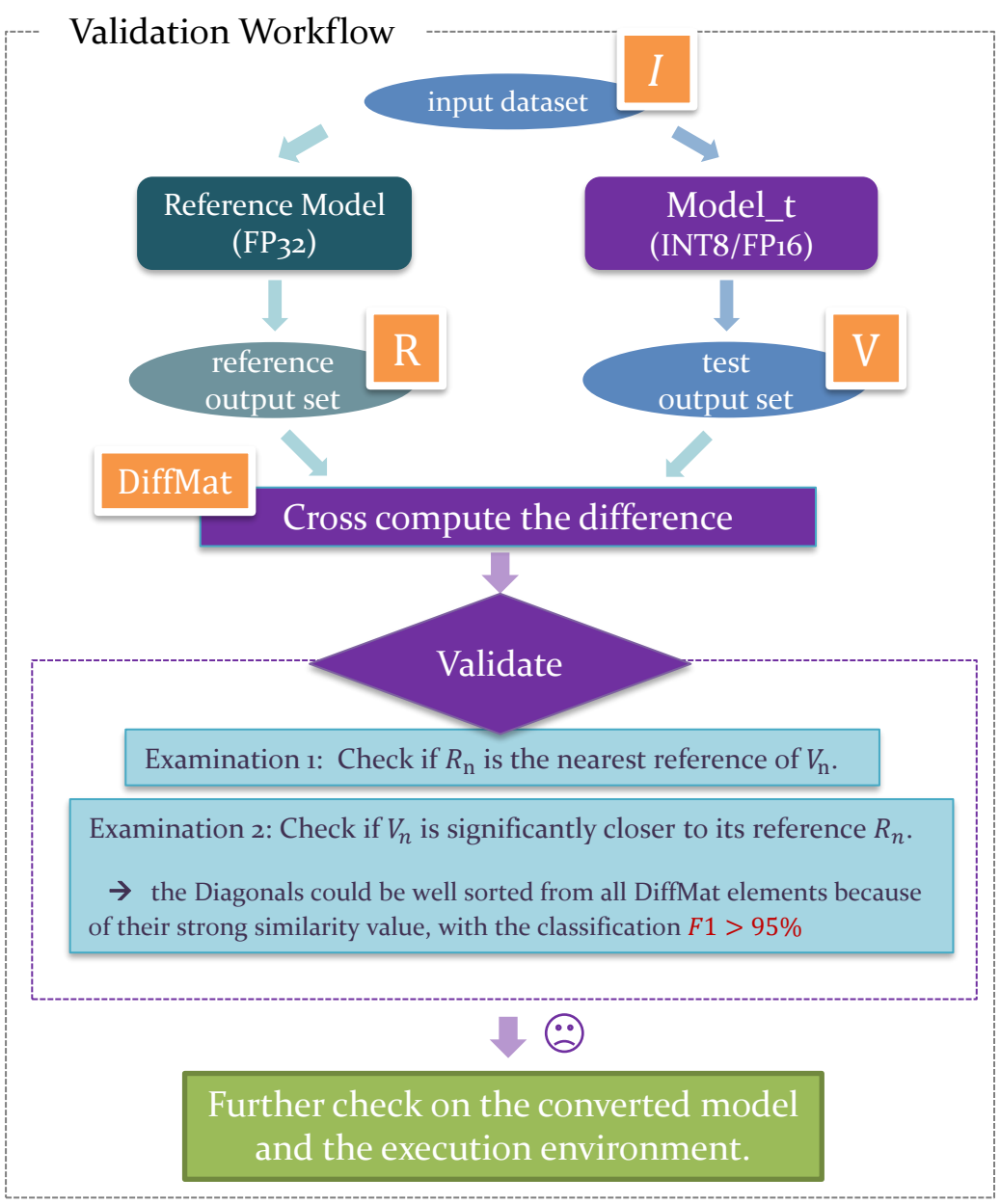
$Recall\ R = \frac{TP}{TP+FN}$

$F1 = \frac{2PR}{P+R}$

If the diffMat classification $F1 > 95\%$, pass the precondition validation test.

The higher the F1-score, the higher the reliability of the model, the execution script and environment.

# Summary




## V.S. accuracy validation?

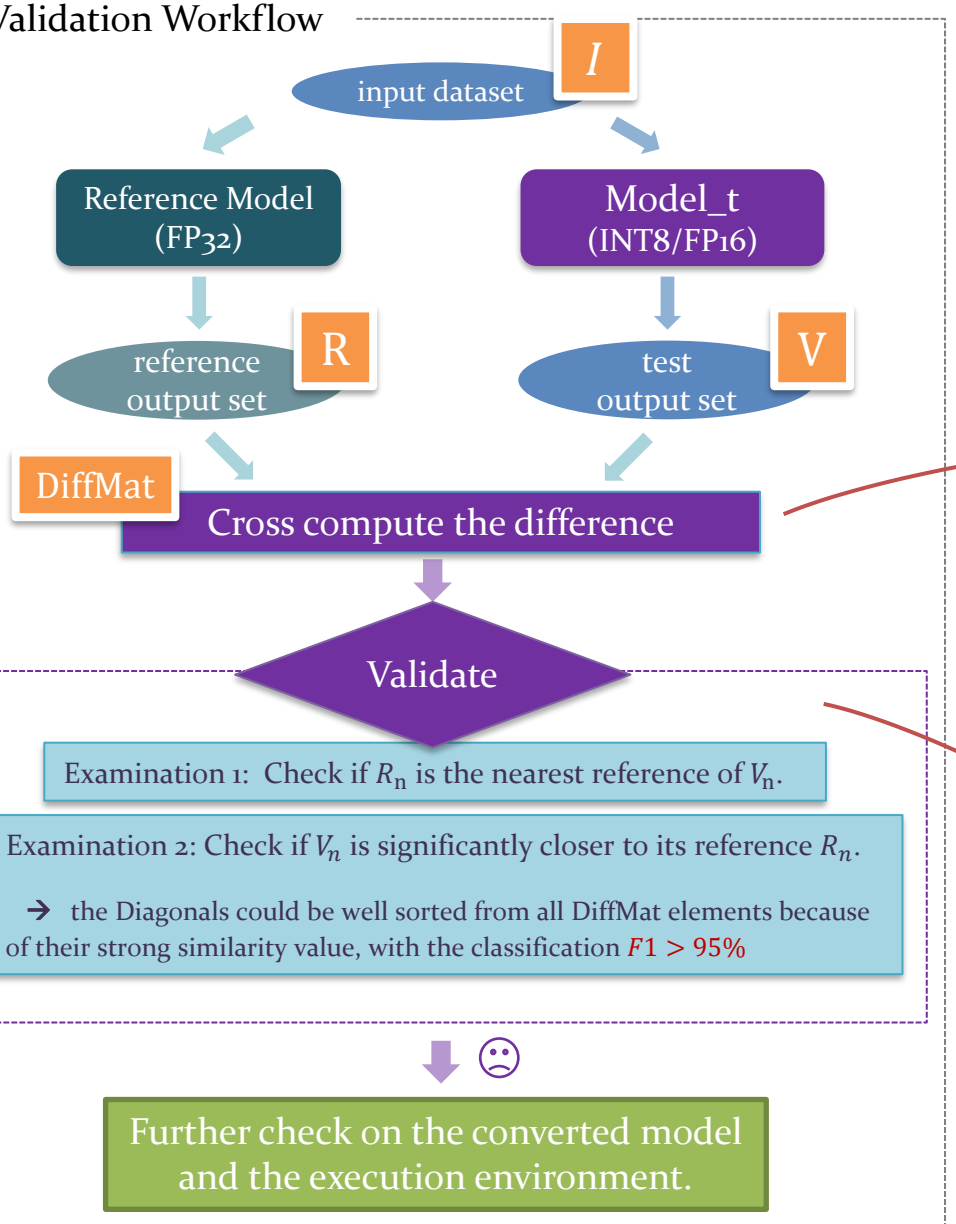
- Utilize more information inside the output feature.
- Help provide more information for the debug/review process.

## V.S. manual inspection?

- Manual check depends on the reviewer's capability and scrutiny. The review quality and speed will fluctuate accordingly.
- A prior unified validation process can make the later debug/review more efficient.

# Appendix

## Validation Workflow




| Step | Test procedure | Expected result |
|---|---|---|
| 1 | Number the data in Test Dataset from 1 to 1000. | Each test data is denoted as I(n), where n ∈ [1, 1000]. |
| 2 | Run Reference Model with Test Dataset on a PC. | The output dataset of PC is obtained. Each data inside is in tensor form, denoted as R(n) according to its input I(n). |
| 3 | Run Model_t with Test Dataset on DUT. | The output dataset of DUT is obtained. |
| 4 | Convert each DUT output data into a tensor, with the shape identical to the shape of R(n). | The converted DUT output data is in tensor form, denoted as V(n) according to its input I(n) in step 3. |

| Step | Test procedure | Expected result |
|---|---|---|
| 5 | Calculate the difference between a given V(n) and each R(m), m ∈ [1, 1000].<br>Note: Euclidean distance is recommended as the difference function. | A difference value set of the given V(n) is obtained. |
| 6 | Repeat step 5 until each V(n) is calculated. | A two-dimensional difference matrix is obtained, denoted as DiffMat. Each element DiffMat[m, n] represents the difference value between R(m) and V(n), where n, m ∈ [1, 1000]. |

| Step | Test procedure | Expected result |
|---|---|---|
| 7 | Count the diagonal elements that are the minimum of their own row, and calculate the proportion of minimum diagonal elements in all DiffMat[n, n]. | The proportion of minimum diagonal elements in all DiffMat[n, n] should be greater than [99%]. |
| 8 | Classify the DiffMat elements into two classes based on their difference values. Label the elements with strong similarity as Positive and the others as Negative. | All elements in DiffMat are sorted and labelled. |
| 9 | Take the Positive diagonal elements as True Positive instances, and calculate the F1-score of the classification. Denote the result as Fc%. (The information and utility preserved in Model_t's output are similar to those preserved in Reference Model's output, under the confidence of Fc%.) | If Fc% is not lower than [95%], it is considered that Model_t and Reference Model have a strong similarity, and Model_t can be validated as Test Model. |