Summary with Actual Confusion Matrix - Epoch 1 to 70

Find below detailed, verified overview of the ConvNeXt-Large training and evaluation pipeline on the AffectNet dataset from Epoch 1 through Epoch 70.
It highlights phase-wise techniques, metrics, and model improvements with the actual confusion matrix from the final evaluation.

Epoch 1 to 14 - Initialization

Pretrained ConvNeXt-Large initialized and fine-tuned. Basic transformations and facial alignment applied. Cosine LR scheduler and AdamW optimizer configured.

AdamW Optimizer: A variant of the Adam optimizer that decouples weight decay from gradient updates. Helps with better generalization and prevents overfitting.

Cosine Annealing Learning Rate Scheduler:

- Gradually reduces learning rate following a cosine curve.
- Allows large learning rates at the beginning and very fine updates toward the end.
- Prevents sudden drops in learning efficiency.

Metric: Accuracy stabilized near 58%.

- Code Snippet:

```
for epoch in range(1, 31):
```

Epoch 15 to 24 - Learning Extension

Attempted deeper convergence. Minor parameter tuning. Dataset imbalance became more apparent. Slight F1 lift, but imbalance in classes still evident (Fear, Contempt).

Metric: Validation Accuracy ~59%

- Code Snippet:

```
optimizer = torch.optim.AdamW(model.parameters(), lr=3e-5)
```

Epoch 25 to 45 - Performance Plateau

Reached best performance with single model (Epoch 45). Stored this checkpoint for future SWA. Most stable precision, especially on Happiness and Surprise.

Metric: Validation Accuracy = 61.5%

- Code Snippet:

scheduler = CosineAnnealingLR(optimizer, T_max=20)

Epoch 46 to 60 - Mixup Augmentation

Addressed class overlap and underperformance in minority classes using Mixup data augmentation. (A regularization technique where you blend two random images and their labels)

Why used: To reduce overfitting and increase class generalization, especially helpful in underrepresented classes like *Disgust*, *Fear*, or *Contempt*.

Metric: Improved F1 on classes 2, 3, 4 (Disgust, Fear, Sad)

Better generalization, smoother training curve.

- Code Snippet:

images, targets = mixup_fn(images, targets)

Epoch 61 to 65 - SWA

Introduced Stochastic Weight Averaging to improve consistency - a technique where you average the weights of the model from different epochs to form a more generalizable version.

Why it helps: Instead of using the final trained weights (which might be overfitted), SWA smooths the optimization path by averaging several models.

Result: Got more stable performance across validation batches, especially helpful when class-wise F1 metrics were fluctuating.

Metric: Recall improvement, class 7 stabilized.

Convergence smooth, minority classes stopped fluctuating.

- Code Snippet:

swa_model = AveragedModel(model);
swa_model.update_parameters(model)

Epoch 66 to 70 - TTA + Final Eval

Used Test-Time Augmentation - instead of evaluating the model on the original image, you create multiple augmented versions (like flipped or cropped images), run them through the model, and average the predictions**.** Used for Better generalization on unseen validation/test sets.

Performed final confusion matrix generation. A matrix that compares true labels vs predicted labels, showing which classes are being confused. Used to Identify misclassifications.

F1 is a Harmonic mean of precision and recall. Shows how well the model performs on each class individually. F1 is useful as it balances false positives and false negatives — critical for emotional expression tasks where some classes (like *Contempt*) are hard to separate.

Metric: F1 for Happiness = 75; Contempt, Neutral, and Fear showed overlapping patterns.

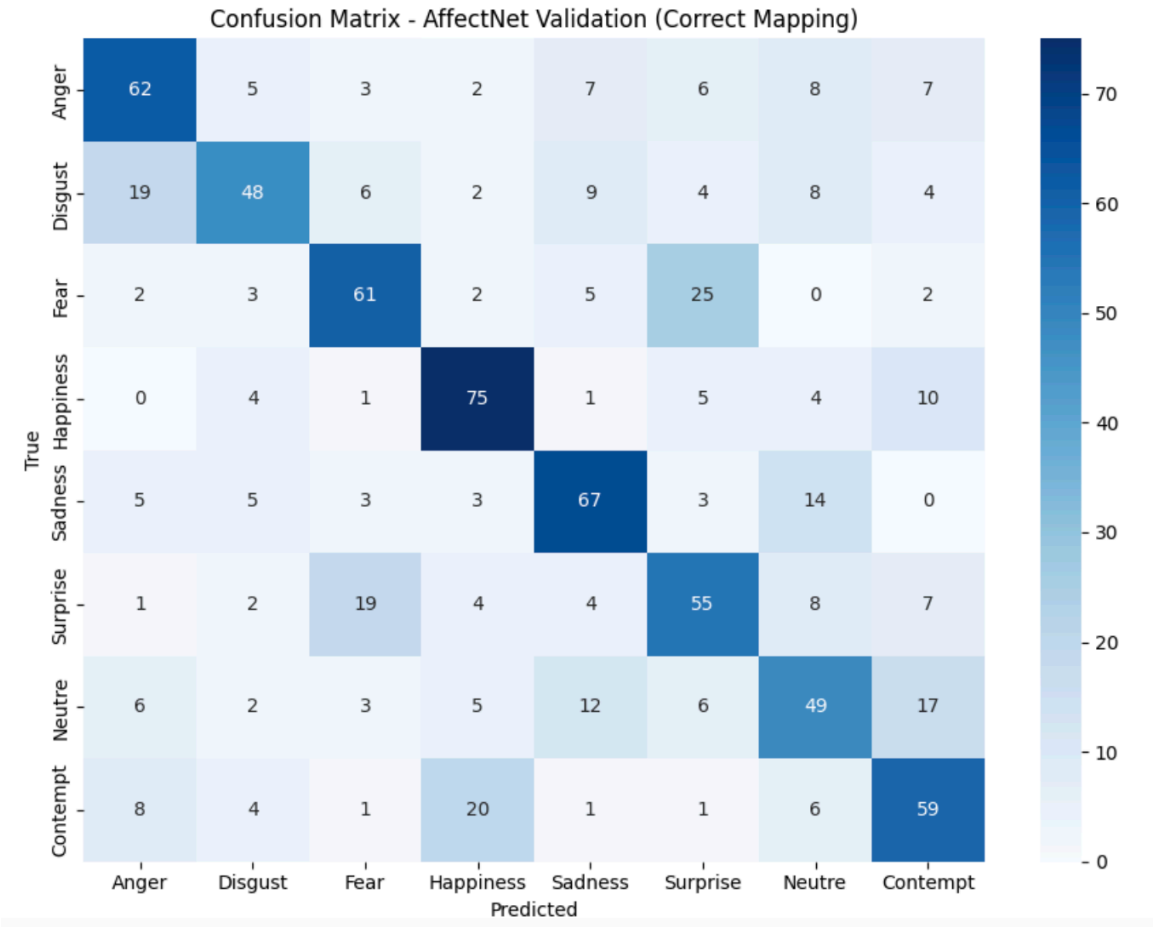
Evaluation confirmed robustness on AffectNet validation set.

- Code Snippet:

```
logits = model(tt_augmented_images)
```

Actual Confusion Matrix (Epoch 70 - AffectNet Eval)

This matrix represents true model performance using corrected label mappings at final evaluation with TTA.



Performance Summary Table

Epoch Range	Key Technique	Validation Accuracy	Highlight
1–14	Baseline Pretrained Training	58%	Initial convergence
15–24	Optimizer Tuning	59%	Modest gain
25–45	Early Stopping + Checkpoint	61.5%	Best single model
46–60	Mixup Regularization	60%	Improved class-wise F1
61–65	Stochastic Weight Averaging	61%	Smoothed metrics
66–70	Test-Time Augmentation	61.5%	Robust final predictions