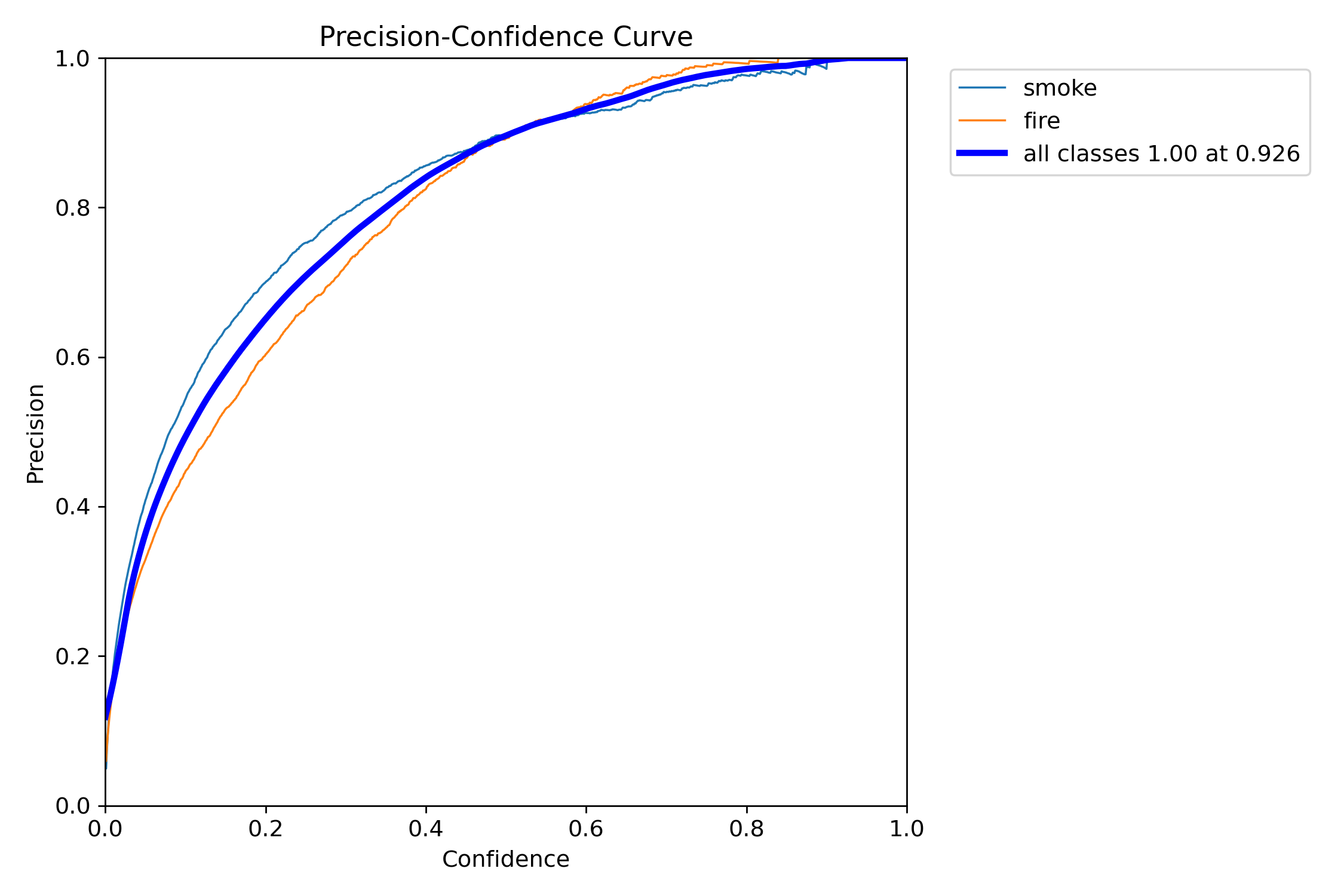
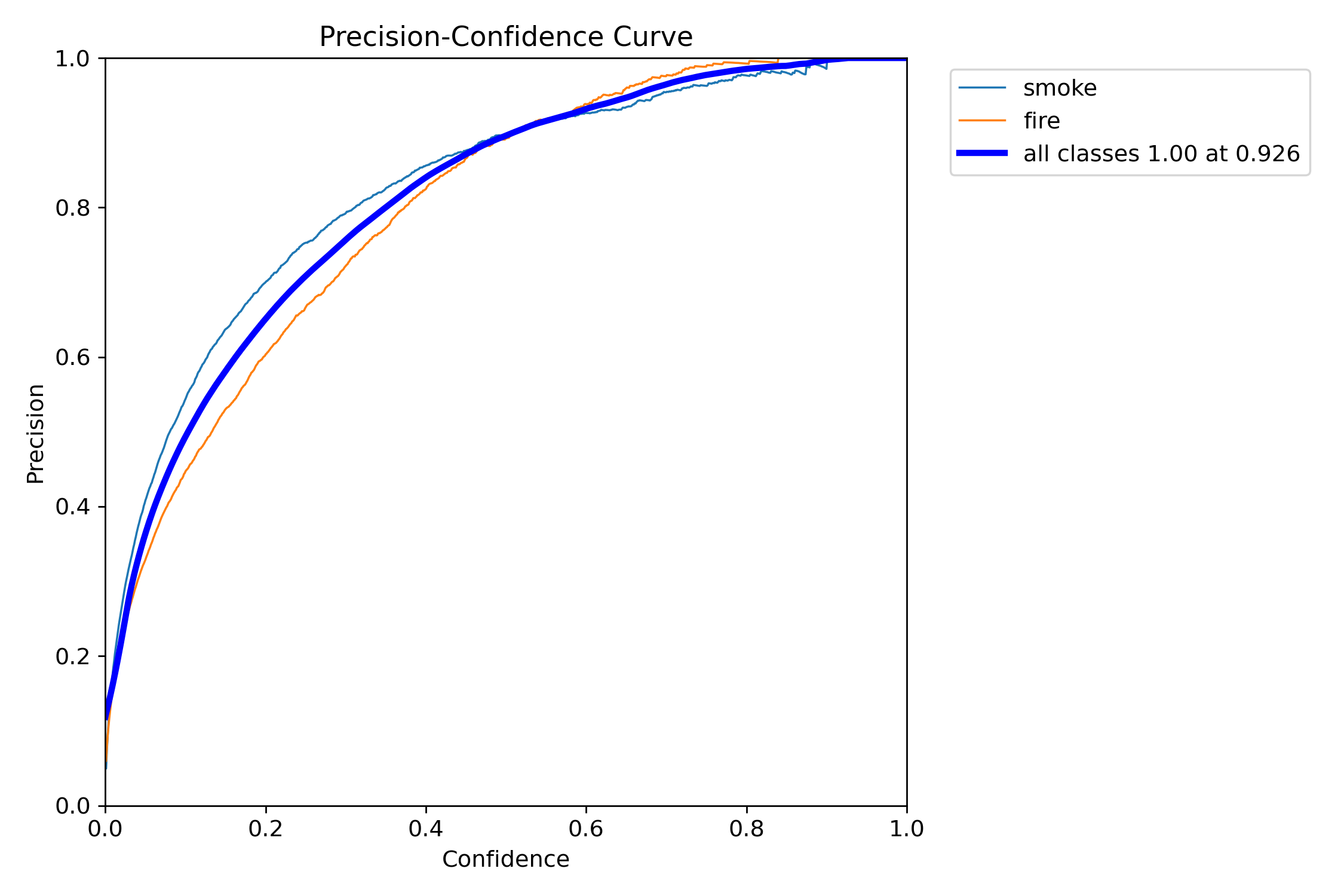


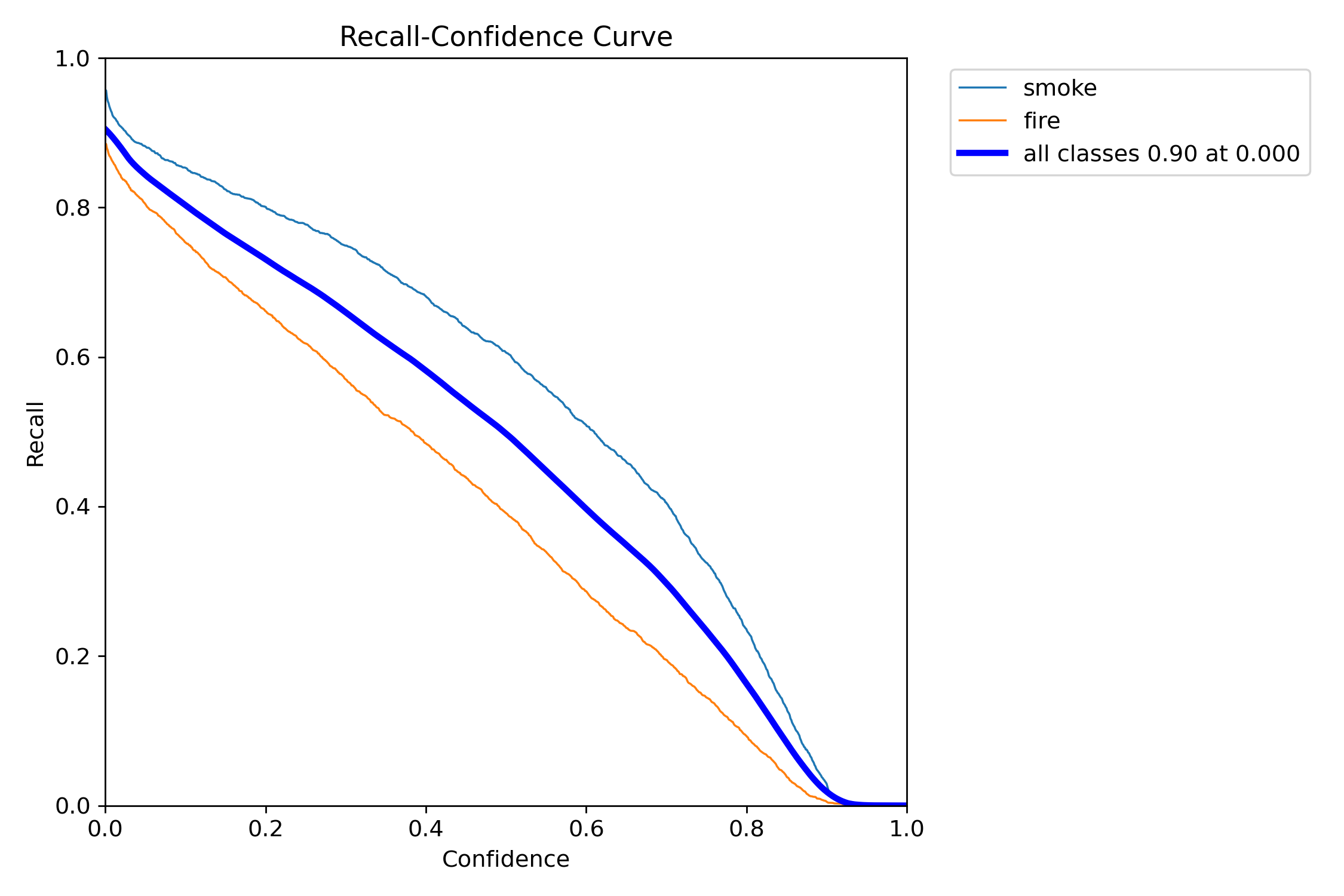
The graph provided is titled “F1-Confidence Curve” and it shows the relationship between the F1 score and the confidence level for detecting “smoke”, “fire”, and “all classes 0.70 at 0.286”. The F1 score is a measure of a model’s accuracy, considering both precision and recall. The confidence level represents the model’s certainty in its predictions. The graph helps to understand the model’s performance at various confidence levels for different classes. By adjusting the confidence threshold, we can potentially improve the F1 score and thus the overall performance of the model.



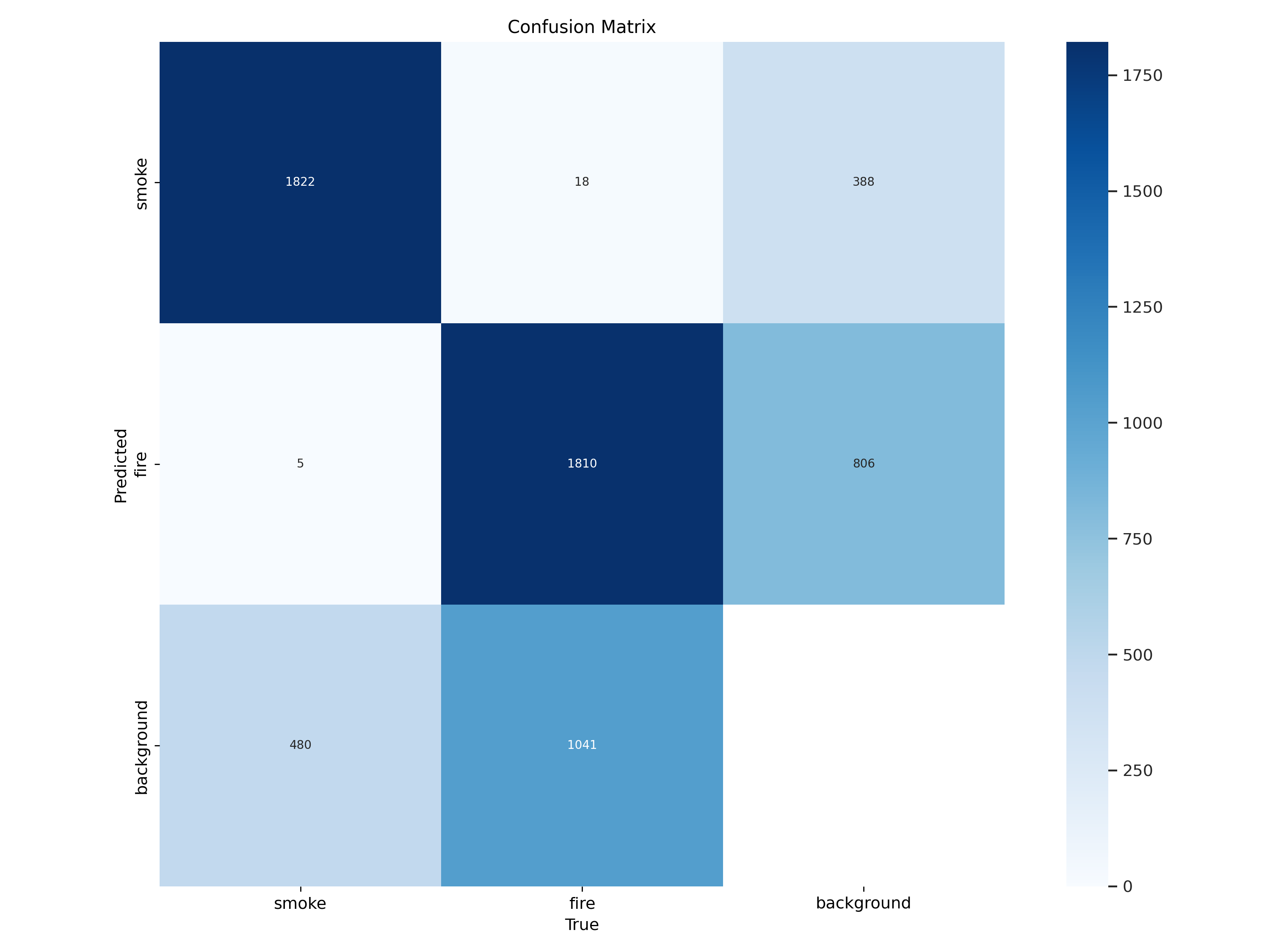
The above graph is a Precision-Confidence Curve graph. It shows the precision of a classification model in detecting “smoke” and “fire” at various confidence levels. As the confidence level increases, the precision of detection also increases. The graph indicates that all classes achieve a precision of 1.00 at a confidence level of 0.926. This graph helps in understanding the model’s performance and the trade-off between confidence and precision.



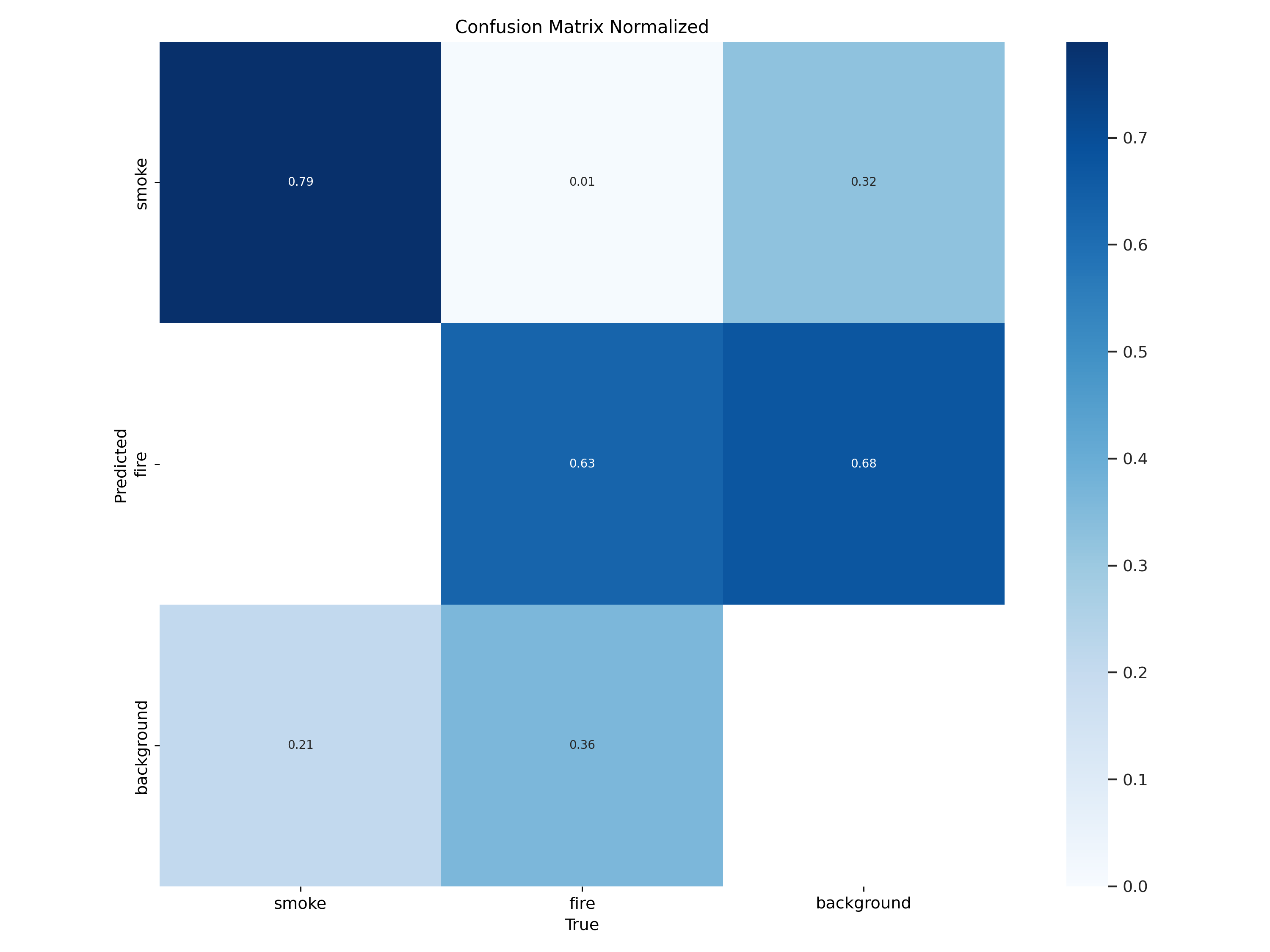
This is a Precision-Confidence Curve graph. It shows the precision of a classification model in identifying “smoke” and “fire” at various confidence levels. As the confidence level increases, the precision of detection also increases. The graph indicates that all classes achieve a precision of 1.00 at a confidence level of 0.926. This graph helps in understanding the model’s performance and the trade-off between confidence and precision.



This is a Recall-Confidence Curve graph. It shows the recall of a classification model in identifying “smoke” and “fire” at various confidence levels. As the confidence level increases, the recall of detection also increases. This graph helps in understanding the model’s performance and the trade-off between confidence and recall. Recall is the number of correct positive results divided by the number of positive results that should have been returned. The “Confidence” on the x-axis represents the model’s confidence in its predictions. A higher confidence means that the model is more certain about its prediction. This type of graph is particularly useful in scenarios where we want to understand the trade-off between the confidence of the model and the recall of its predictions. By adjusting the confidence threshold, we can potentially improve the recall and thus the overall performance of the model.



It visualizes the performance of the model in distinguishing between “smoke”, “fire”, and “background”. The matrix has actual classes labeled on the y-axis and predicted classes on the x-axis. Each cell in the grid represents the count of instances for the corresponding combination of actual and predicted classes. The color intensity correlates with the count number, with darker shades representing higher counts. The model has accurately identified most instances of smoke and fire, but there were some misclassifications where smoke was classified as background and vice versa. A small number of fire instances were also misclassified as smoke or background. This matrix helps in understanding the model’s accuracy and errors.



It visualizes the performance of the model in distinguishing between “smoke”, “fire”, and “background”. The matrix has actual classes labeled on the y-axis and predicted classes on the x-axis. Each cell contains values representing the proportion of correct and incorrect predictions normalized. The diagonal cells represent correct predictions for smoke, fire, and background respectively. The off-diagonal cells indicate errors made by the model in classification. A color scale on the right indicates the values’ range from 0 to 0.7. This matrix helps in understanding the model’s accuracy and errors.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| epoch | train/box\_loss | train/cls\_loss | train/dfl\_loss | metrics/precision(B) | metrics/recall(B) | metrics/mAP50(B) | metrics/mAP50-95(B) | val/box\_loss | val/cls\_loss | val/dfl\_loss | lr/pg0 | lr/pg1 | lr/pg2 |
| 1 | 1.7619 | 2.0295 | 1.6275 | 0.39459 | 0.36999 | 0.31356 | 0.12605 | 1.6835 | 8.6667 | 1.4127 | 0.00055515 | 0.00055515 | 0.00055515 |
| 2 | 1.8006 | 1.9392 | 1.6338 | 0.42222 | 0.35086 | 0.3075 | 0.13086 | 1.7313 | 5.7817 | 1.4399 | 0.0010668 | 0.0010668 | 0.0010668 |
| 3 | 1.7496 | 1.8456 | 1.5858 | 0.49708 | 0.36571 | 0.37928 | 0.1709 | 1.6042 | 2.6514 | 1.3171 | 0.0015345 | 0.0015345 | 0.0015345 |
| 4 | 1.7024 | 1.7713 | 1.5526 | 0.48301 | 0.44944 | 0.43268 | 0.20807 | 1.5153 | 4.607 | 1.2918 | 0.001469 | 0.001469 | 0.001469 |
| 5 | 1.6402 | 1.6582 | 1.5178 | 0.53774 | 0.50466 | 0.49831 | 0.23768 | 1.489 | 3.7102 | 1.2445 | 0.001469 | 0.001469 | 0.001469 |
| 6 | 1.6089 | 1.5899 | 1.4937 | 0.57203 | 0.49728 | 0.52158 | 0.24661 | 1.4376 | 3.1325 | 1.2049 | 0.0014029 | 0.0014029 | 0.0014029 |
| 7 | 1.573 | 1.5298 | 1.4691 | 0.61415 | 0.4906 | 0.53025 | 0.27322 | 1.4252 | 2.7005 | 1.1986 | 0.0013369 | 0.0013369 | 0.0013369 |
| 8 | 1.5402 | 1.4679 | 1.4461 | 0.63034 | 0.55126 | 0.58845 | 0.30163 | 1.3465 | 3.3299 | 1.161 | 0.0012709 | 0.0012709 | 0.0012709 |
| 9 | 1.5182 | 1.4431 | 1.4313 | 0.63153 | 0.57025 | 0.60415 | 0.31998 | 1.3558 | 3.1333 | 1.1493 | 0.0012049 | 0.0012049 | 0.0012049 |
| 10 | 1.5046 | 1.3864 | 1.4198 | 0.65039 | 0.59017 | 0.62853 | 0.33079 | 1.3174 | 2.4674 | 1.1274 | 0.0011389 | 0.0011389 | 0.0011389 |
| 11 | 1.4788 | 1.3604 | 1.4083 | 0.66498 | 0.58028 | 0.63199 | 0.33741 | 1.3239 | 1.8368 | 1.1321 | 0.0010729 | 0.0010729 | 0.0010729 |
| 12 | 1.4548 | 1.3337 | 1.3941 | 0.66987 | 0.5911 | 0.65075 | 0.35392 | 1.2741 | 2.4773 | 1.1119 | 0.0010069 | 0.0010069 | 0.0010069 |
| 13 | 1.443 | 1.3125 | 1.3888 | 0.68058 | 0.59853 | 0.65937 | 0.35686 | 1.2682 | 2.35 | 1.0986 | 0.00094085 | 0.00094085 | 0.00094085 |
| 14 | 1.4321 | 1.2845 | 1.375 | 0.69303 | 0.62435 | 0.67504 | 0.37021 | 1.2649 | 2.1989 | 1.0985 | 0.00087484 | 0.00087484 | 0.00087484 |
| 15 | 1.4135 | 1.2408 | 1.3635 | 0.72347 | 0.60788 | 0.68446 | 0.377 | 1.2563 | 1.7648 | 1.095 | 0.00080883 | 0.00080883 | 0.00080883 |
| 16 | 1.4632 | 1.2179 | 1.377 | 0.71325 | 0.6191 | 0.67686 | 0.37812 | 1.2479 | 1.6649 | 1.0902 | 0.00074282 | 0.00074282 | 0.00074282 |
| 17 | 1.448 | 1.1887 | 1.3641 | 0.69487 | 0.62694 | 0.6838 | 0.38057 | 1.2299 | 1.9785 | 1.0786 | 0.0006768 | 0.0006768 | 0.0006768 |
| 18 | 1.4243 | 1.1458 | 1.3502 | 0.70073 | 0.63616 | 0.69579 | 0.38632 | 1.2122 | 1.7667 | 1.0755 | 0.00061079 | 0.00061079 | 0.00061079 |
| 19 | 1.3984 | 1.1095 | 1.3378 | 0.7199 | 0.64287 | 0.70893 | 0.39924 | 1.2129 | 1.6826 | 1.07 | 0.00054478 | 0.00054478 | 0.00054478 |
| 20 | 1.3809 | 1.085 | 1.318 | 0.72035 | 0.64753 | 0.71564 | 0.40461 | 1.2 | 1.4659 | 1.0614 | 0.00047876 | 0.00047876 | 0.00047876 |
| 21 | 1.3678 | 1.0597 | 1.3147 | 0.73848 | 0.65107 | 0.72216 | 0.41204 | 1.1917 | 1.5733 | 1.0588 | 0.00041275 | 0.00041275 | 0.00041275 |
| 22 | 1.3475 | 1.0199 | 1.308 | 0.73611 | 0.65844 | 0.7263 | 0.4166 | 1.193 | 1.4493 | 1.0629 | 0.00034674 | 0.00034674 | 0.00034674 |
| 23 | 1.3237 | 0.99141 | 1.2864 | 0.71942 | 0.67224 | 0.73069 | 0.41917 | 1.2038 | 1.3606 | 1.06 | 0.00028072 | 0.00028072 | 0.00028072 |
| 24 | 1.3109 | 0.97404 | 1.2807 | 0.74594 | 0.66673 | 0.73673 | 0.42485 | 1.1819 | 1.297 | 1.0498 | 0.00021471 | 0.00021471 | 0.00021471 |
| 25 | 1.2934 | 0.95162 | 1.2672 | 0.74395 | 0.67223 | 0.73987 | 0.42779 | 1.1853 | 1.2659 | 1.059 | 0.0001487 | 0.0001487 | 0.0001487 |

This is a record of a our YOLOV9 model’s performance over 25 epochs. It tracks various losses during training (train/box\_loss, train/cls\_loss, train/dfl\_loss) and validation (val/box\_loss, val/cls\_loss, val/dfl\_loss). These losses decrease over epochs, indicating the model is learning effectively. The table also monitors precision, recall, and mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds (metrics/precision(B), metrics/recall(B), metrics/mAP50(B), metrics/mAP50-95(B)). These metrics increase over time, suggesting improved model accuracy. Lastly, the learning rates (lr/pg0, lr/pg1, lr/pg2) decrease, which is typical as training progresses to allow the model to make finer adjustments. This table provides a comprehensive view of the model’s learning trajectory and performance.

1. Training Losses: These values represent how well the model is performing on the training data. The train/box\_loss is related to the bounding box prediction, train/cls\_loss is the classification loss, and train/dfl\_loss is the deformable convolution layer loss. A decrease in these values over epochs indicates the model is improving its predictions on the training data.
2. Validation Losses: These are similar to the training losses but calculated on the validation data. A decrease in val/box\_loss, val/cls\_loss, and val/dfl\_loss suggests the model is generalizing well and not overfitting the training data.
3. Metrics: These values measure the model’s predictive accuracy. metrics/precision(B) is the ratio of true positives to the sum of true and false positives. metrics/recall(B) is the ratio of true positives to the sum of true positives and false negatives. metrics/mAP50(B) and metrics/mAP50-95(B) are the mean Average Precision at different Intersection over Union (IoU) thresholds. An increase in these metrics over time indicates the model’s predictive accuracy is improving.
4. Learning Rate: The lr/pg0, lr/pg1, and lr/pg2 values represent the learning rate for different parameter groups in the model. The learning rate controls the step size in model parameter updates. A decreasing learning rate allows the model to make smaller, more precise adjustments in later epochs.

A screenshot of a video

Description automatically generatedA screenshot of a video

Description automatically generatedA screenshot of a video

Description automatically generatedA screenshot of a video

Description automatically generatedA screenshot of a video

Description automatically generatedA screenshot of a video

Description automatically generated

val: New cache created: /content/test/labels.cache

Class Images Instances Box(P R mAP50 mAP50-95): 100% 269/269 [01:04<00:00, 4.16it/s]

all 4298 5176 0.741 0.673 0.739 0.428

smoke 4298 2307 0.781 0.759 0.806 0.5

fire 4298 2869 0.701 0.586 0.673 0.356

Speed: 0.1ms preprocess, 7.7ms inference, 0.0ms loss, 1.9ms postprocess per image

Results saved to runs/detect/val

