**Comparação dos 4 Modelos Testados**

| **Configuração** | **MobileNetV2 (1 camada, 256 neurônios)** | **MobileNetV2 (2 camadas, 512 neurônios)** | **MobileNetV1 (2 camadas, 512 neurônios)** | **MobileNetV1 (rede menor)** |
| --- | --- | --- | --- | --- |
| **Threshold Médio** | **0.599994** | **0.599994** | **0.599994** | **0.599994** |
| **Acurácia** | **94.30%** | **94.40%** | **94.09%** | **94.61%** |
| **Precisão** | **94.26%** | **95.24%** | **94.94%** | **94.55%** |
| **Recall** | **95.18%** | **94.30%** | **94.01%** | **95.48%** |
| **F1-Score** | **94.72%** | **94.76%** | **94.47%** | **95.01%** |
| **ROC-AUC** | **97.82%** | **97.72%** | **97.44%** | **97.31%** |
| **Verdadeiros Negativos** | **817** | **828** | **825** | **820** |
| **Falsos Positivos** | **59** | **48** | **51** | **56** |
| **Falsos Negativos** | **49** | **58** | **61** | **46** |
| **Verdadeiros Positivos** | **969** | **960** | **957** | **972** |

**📌 Análise Comparativa**

✅ **MobileNetV1 (rede menor) teve a melhor acurácia (94.61%)**, superando os outros três modelos.  
✅ **MobileNetV1 (rede menor) teve o maior recall (95.48%)**, indicando que ele capturou mais positivos corretamente.  
✅ **MobileNetV2 (1 camada, 256 neurônios) teve o melhor ROC-AUC (97.82%)**, o que sugere melhor separação entre classes.  
✅ **MobileNetV2 (2 camadas, 512 neurônios) teve a maior precisão (95.24%)**, reduzindo falsos positivos.  
✅ **O MobileNetV1 (rede menor) obteve o melhor F1-Score (95.01%)**, indicando o melhor equilíbrio entre precisão e recall.  
✅ **O MobileNetV1 (rede menor) também teve o menor número de falsos negativos (46)**, o que significa menos amostras positivas classificadas incorretamente como negativas.

**Qual Modelo Escolher?**

🔹 **Se quiser um modelo mais eficiente e que mantém bom desempenho** → **MobileNetV1 (rede menor) é uma ótima escolha, pois conseguiu reduzir falsos negativos e melhorar o recall.**  
🔹 **Se quiser um modelo com melhor separação entre classes** → **MobileNetV2 (1 camada, 256 neurônios) tem o melhor ROC-AUC.**  
🔹 **Se quiser um modelo com menos falsos positivos** → **MobileNetV2 (2 camadas, 512 neurônios) tem a maior precisão.**

📌 **Conclusão:** O **MobileNetV1 (rede menor)** mostrou um ótimo equilíbrio entre desempenho e eficiência. Ele conseguiu **melhorar recall, acurácia e F1-Score** sem perder muito em precisão, o que pode ser ideal para aplicações onde minimizar falsos negativos é importante. 🚀🔥

Para uma aplicação de **detecção de obstáculos para pessoas com deficiência visual**, o modelo precisa ser:

✅ **Preciso** → Para minimizar falsos positivos e negativos.  
✅ **Rápido e leve** → Para rodar em um dispositivo móvel em tempo real.  
✅ **Sensível a obstáculos** → Para minimizar falsos negativos (não reconhecer um obstáculo pode ser perigoso).

**📌 Escolha da Melhor Abordagem**

Com base nesses critérios, a **melhor abordagem seria o MobileNetV1 com rede neural menor**. 🚀

🔹 **Motivos para essa escolha:**

1. **Menor tempo de inferência** 🏎️ → MobileNetV1 é mais leve e consome menos processamento, crucial para dispositivos móveis.
2. **Maior Recall (95.48%)** ✅ → Minimiza o risco de obstáculos passarem despercebidos, o que pode ser perigoso para o usuário.
3. **Acurácia e F1-Score competitivos** 📊 → Melhor equilíbrio geral entre acertos, falsos positivos e falsos negativos.
4. **Consumo menor de energia** 🔋 → Essencial para um sistema rodando continuamente em um celular ou dispositivo vestível.
5. **Facilidade de implementação no TensorFlow Lite** 📲 → MobileNet já tem suporte otimizado para Android/Kotlin.

**📌 Justificativa Técnica**

* **Um alto recall (95.48%)** garante que obstáculos sejam corretamente detectados.
* **Acurácia de 94.61%** significa que o modelo tem um desempenho global robusto.
* **Falsos negativos menores (46)** → Isso reduz o risco de não alertar sobre um obstáculo.
* **Falsos positivos ligeiramente maiores** → Mas isso pode ser ajustado com um threshold mais dinâmico.

🔍 **A única desvantagem do MobileNetV1 (rede menor) é que ele pode gerar alguns alertas falsos**, mas para um sistema assistivo, é **melhor alertar sobre um obstáculo inexistente do que não alertar sobre um real**.

**📌 Conclusão**

📢 **O MobileNetV1 com rede menor é a melhor escolha** para uma aplicação **assistiva para deficientes visuais**, pois equilibra **velocidade, consumo de energia e precisão na detecção de obstáculos**.

🎯 **Próximos passos**:  
✅ **Converter o modelo para TensorFlow Lite** para uso em dispositivos móveis.  
✅ **Ajustar o threshold para reduzir falsos positivos** e encontrar o melhor equilíbrio.  
✅ **Otimizar o pipeline de inferência** para reduzir latência no processamento da câmera.

🚀 **Essa abordagem garantirá que o sistema seja rápido, eficiente e seguro!** 🔥📱