

Este documento resume lo hecho hasta el momento en detección y corrección de especularidades:

Para este punto seguí adelante basado en el trabajo de [G. Zimmerman-Moreno and H. Greenspan¹](#).

A partir de las imágenes, define la intensidad y saturación como sigue:

$$I = \frac{R+G+B}{3} \quad S = 1 - \frac{\min(R,G,B)}{I}$$

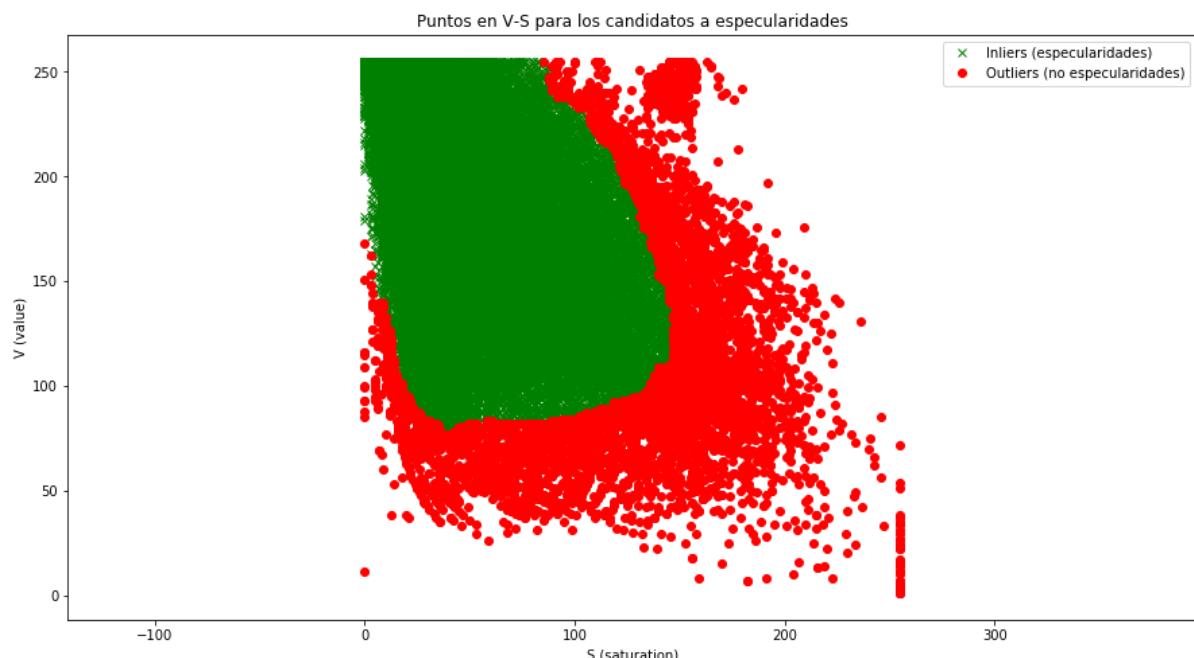
Sobre estos valores aplica un umbral a cada uno, y toma como candidatos a los píxeles que cumplen ambas restricciones. Hice lo mismo, definiendo un umbral diferente en función de las imágenes que tenemos.

Luego, refina estos candidatos calculando el gradiente en I, y aplicando un umbral sobre el resultado. En mi caso, hice lo mismo, pero en vez de refinar el conjunto de candidatos, lo amplié.

Por último el trabajo se toma los píxeles candidatos, los pasa al espacio HSV, y sobre los pares VS entrena un modelo de mezcla de gaussianas para determinar cuáles candidatos son efectivamente especularidades y cuáles no.

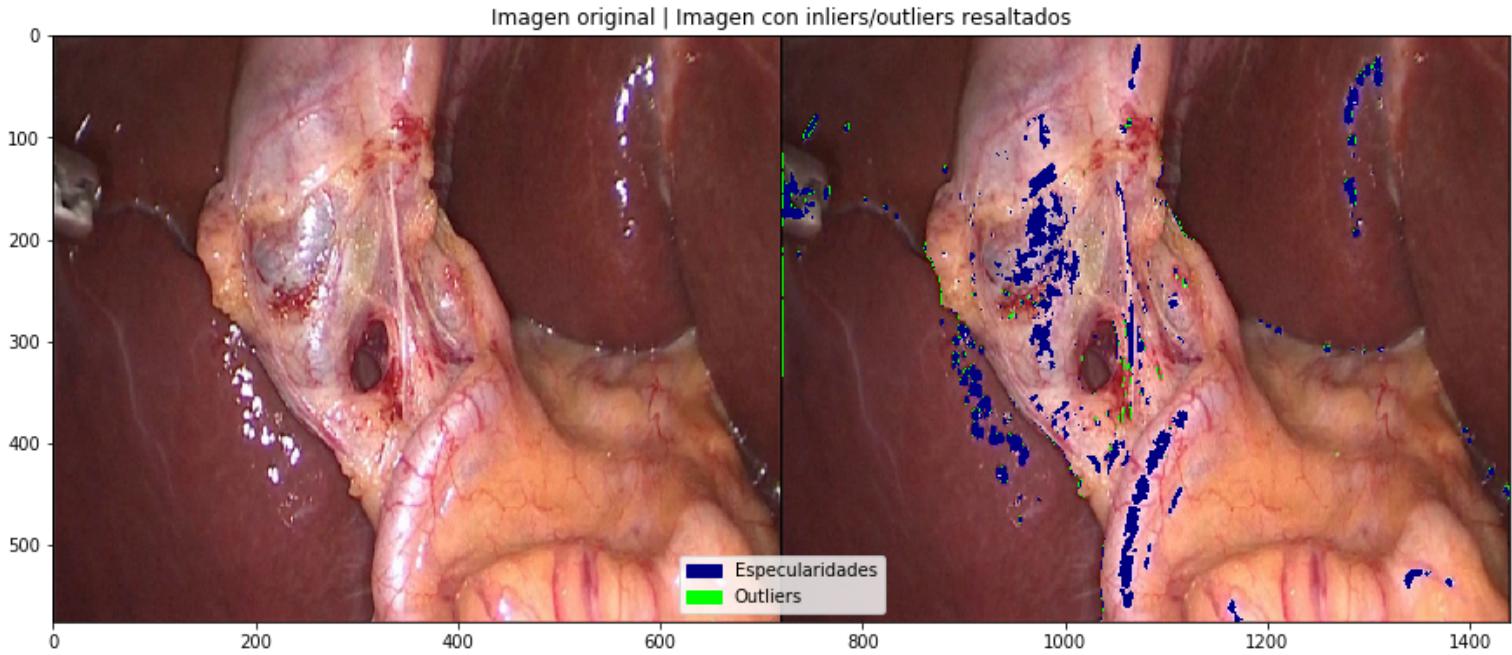
Como los píxeles no están etiquetados, toma la gaussiana que contiene los píxeles con V alto y S bajo como los de las especularidades.

En mi caso intente refinarlo con un enfoque de detección de anomalías, usando IsolationForest, ya que la mayoría de los candidatos efectivamente son especularidades.

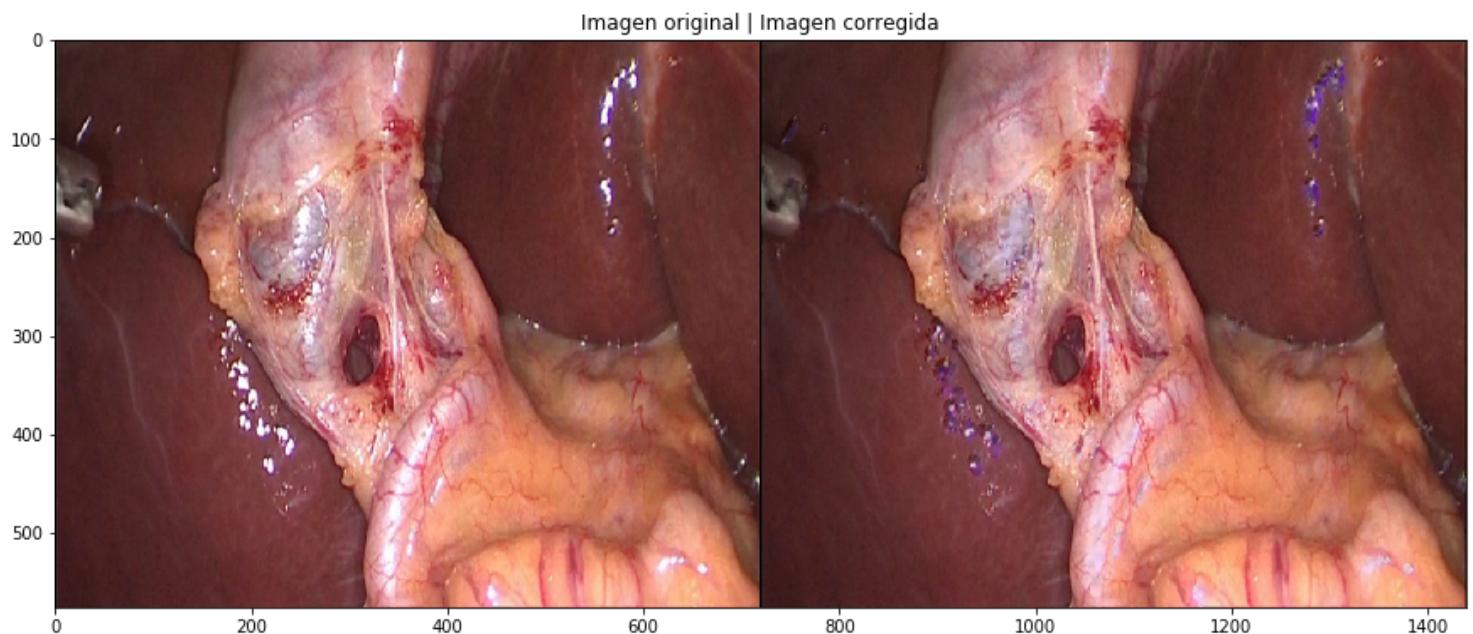


¹Automatic Detection of Specular Reflections in Uterine Cervix Images, 2006

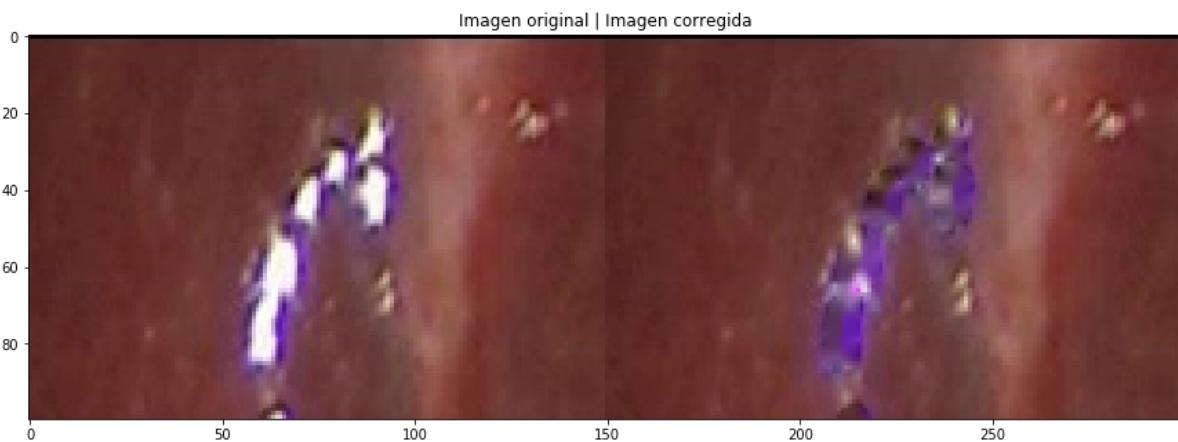
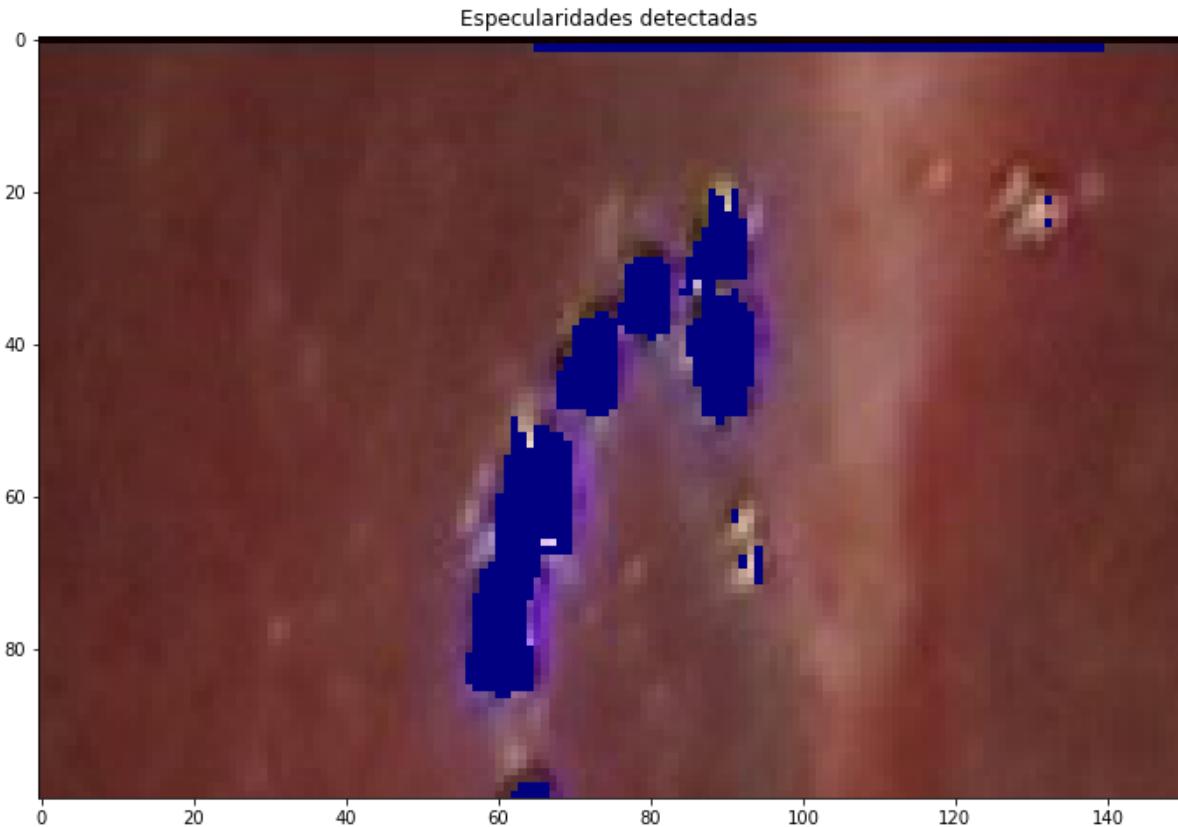
La mayoría de los pixeles no cambió su categoría de especularidades, lo que me hace pensar que quizá este paso sea innecesario, además los que cambian son pequeños, por lo que si los dejamos como especularidades deberían suavizarse durante el inpainting.



Para el inpainting solo probé las dos implementaciones que provee OpenCV. No noté diferencias significativas entre ellas. Un ejemplo de restauración, siguiendo el proceso descrito, es este:

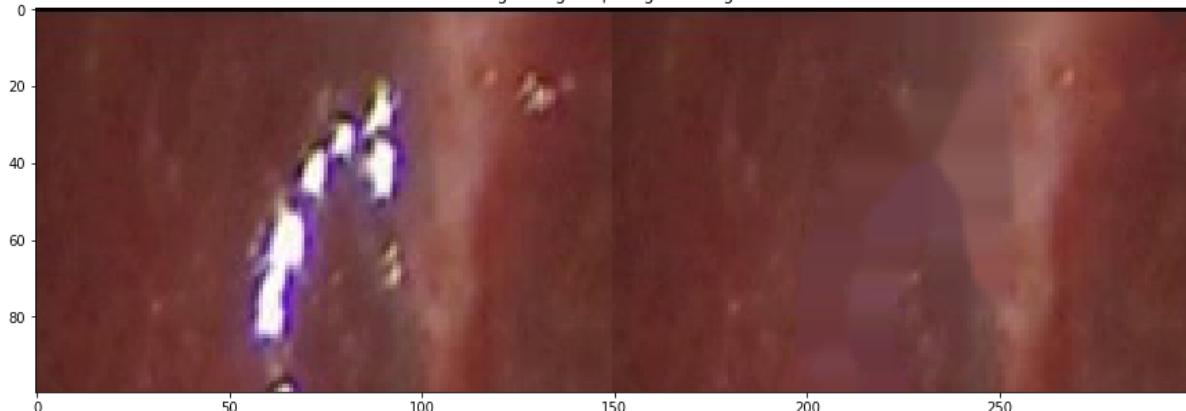


Si bien a primera vista parece un buen resultado, saltan a la vista unas manchas azules/violetas donde habían especularidades. Mirando de cerca, el problema está en que, entorno a las especularidades queda un resplandor azulado que no es detectado como especularidad, y como queda justo en el borde de la especularidad detectada, el inpainting arrastra ese azulado a la especularidad.



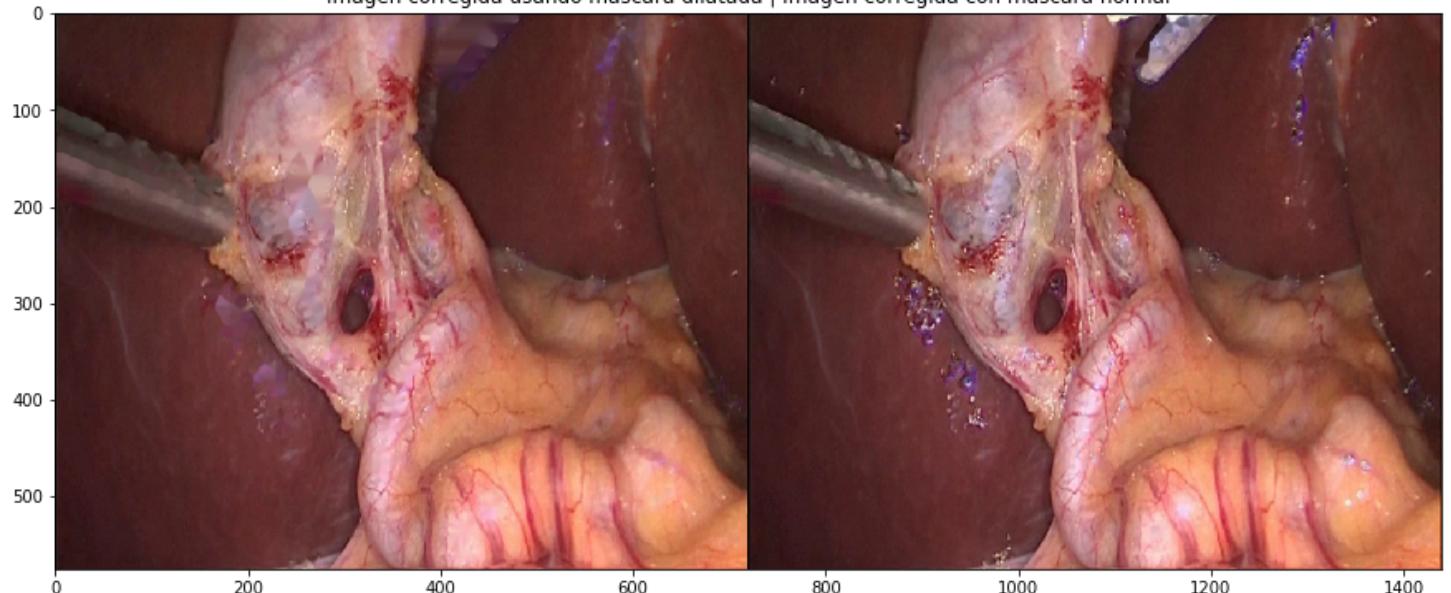
Una solución rápida que se me ocurrió, fue dilatar la máscara de las especulaciones detectadas. Esto debería tapar el azulado y mejorar el inpainting. Lo probé y efectivamente mejoró considerablemente los resultados.

Imagen original | Imagen corregida



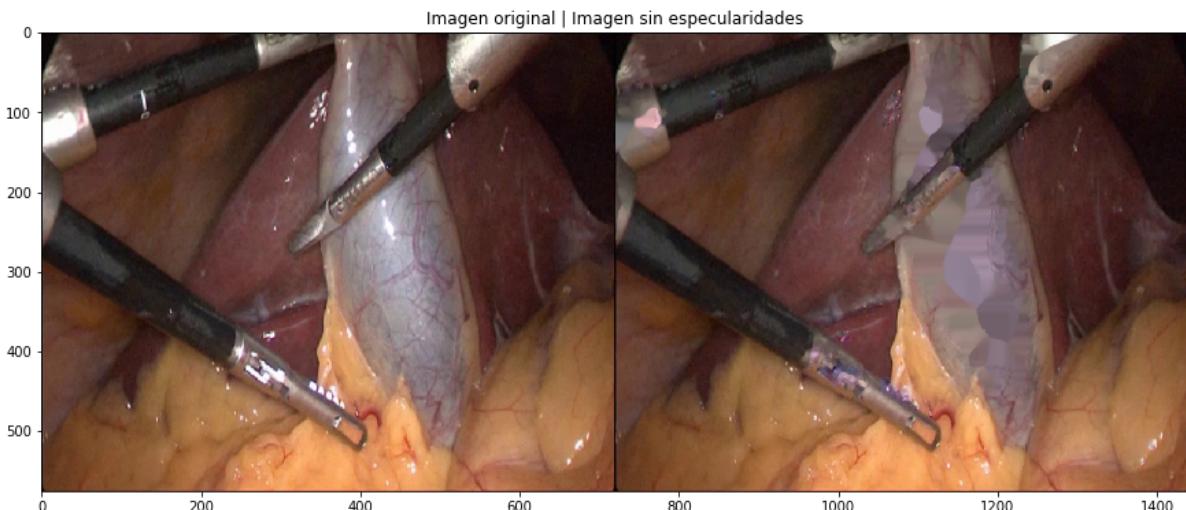
Sin embargo, en las zonas de instrumentos como tienen muchas especularidades, en particular la tijera es casi toda una specularidad, al ampliar la máscara y hacer el inpainting, la tijera se pierde.

Imagen corregida usando mascara dilatada | Imagen corregida con mascara normal



En este punto decidí descartar la dilatación de la máscara, porque es preferible que la reconstrucción pierda calidad a que destruya la tijera.

Con esto, tome una imagen de otro video, con la que no hice ningún ajuste, y aplique todo el pipeline desarrollado para validar el enfoque:

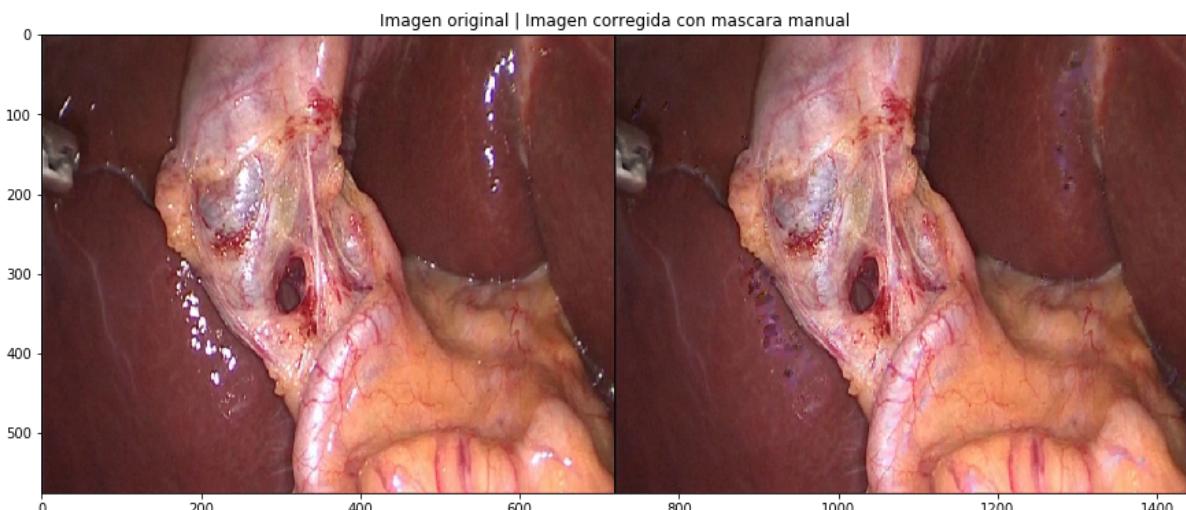


El resultado no es muy bueno. La imagen se distorsiona bastante. En este punto no invertí tiempo en ver cual paso es el que introduce el error, aunque posiblemente sean varios.

Para llegar a este punto, hice varias pruebas intermedias sobre si incluir o no el gradiente, detectar los outliers usando OneClassSVM, y varias combinaciones de parámetros. Lo mejor a lo que llegué fue lo que expuse aquí, y todas las variantes probadas tienen el inconveniente de el resplandor azul, de manera que reconsideré descartar este enfoque de raíz y pasar a segmentar pixeles y clasificarlos en especularidad/no especularidad.

Con esto en mente, genere manualmente una máscara para las especularidades, considerando las partes azuladas como especularidades. Durante el proceso de anotación, que tomó unas cuantas horas, note dos cosas: como el resplandor azulado es como un esfumado, es difícil definir hasta qué punto es resplandor y en cual no. Por otro lado, en algunos puntos claros aislados es difícil determinar si se trata de especularidades o es el color del tejido.

Con esta máscara como ground truth, intente primero que nada hacer el inpainting, ya que en el mejor de los casos un clasificador arribará a esa máscara. El resultado fue el siguiente



Si bien esta imagen no está perfecta, es mejor que la que encontramos segmentando a mano. Con esta motivación comencé a explorar clasificadores para esta tarea. Cómo features tome los que ya había usado: el color en HSV y RGB, la I y S definidas en el trabajo de referencia, y el gradiente sobre I, ya que con estos valores en definitiva se habían logrado resultados.

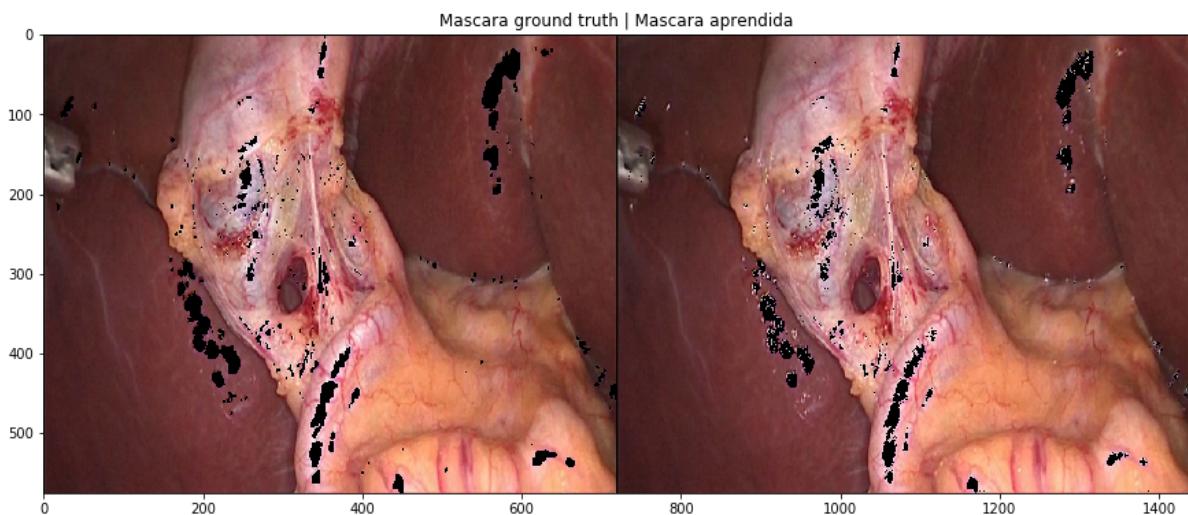
De los píxeles de la imagen, tome el 20% para la validación final y el resto para entrenamiento.

Probé varios clasificadores, y para cada uno de ellos varios hiperparámetros. En cada uno de ellos intente maximizar el promedio de la medida f1 sobre una validación cruzada con 10 folds, tomando como clase positiva a las especularidades.

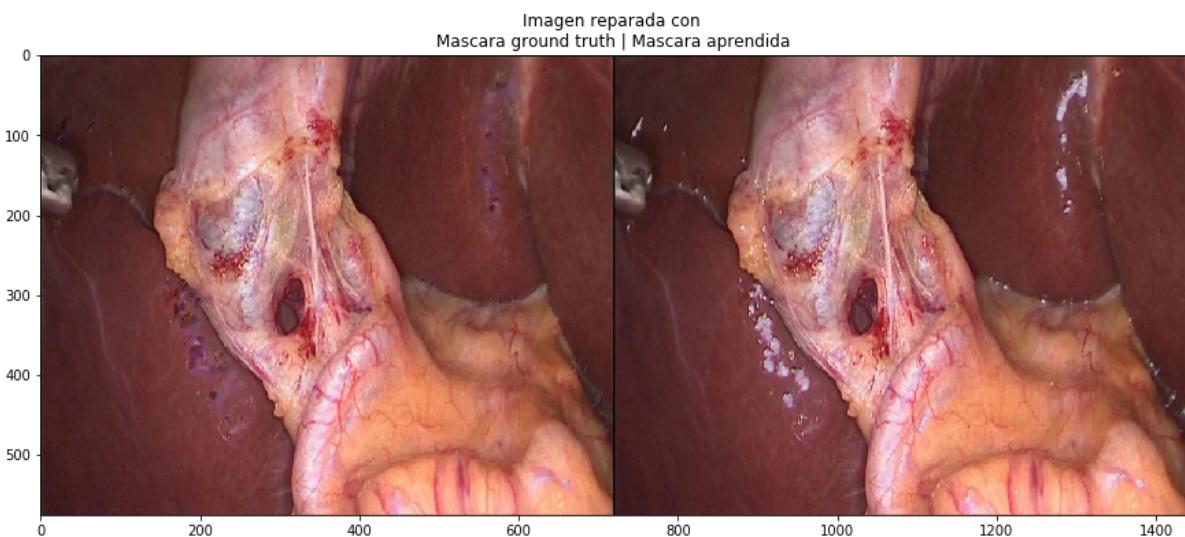
El mejor resultado que obtuve fue con KNN, con $f1 = 0.712 (+/-0.020)$, por lo que proseguí a realizar la evaluación sobre los píxeles que había guardado para testear. El resultado fue el siguiente:

	precision	recall	f1-score	support
0	0.99	1.00	0.99	80406
1	0.81	0.62	0.70	2538
avg / total	0.98	0.98	0.98	82944

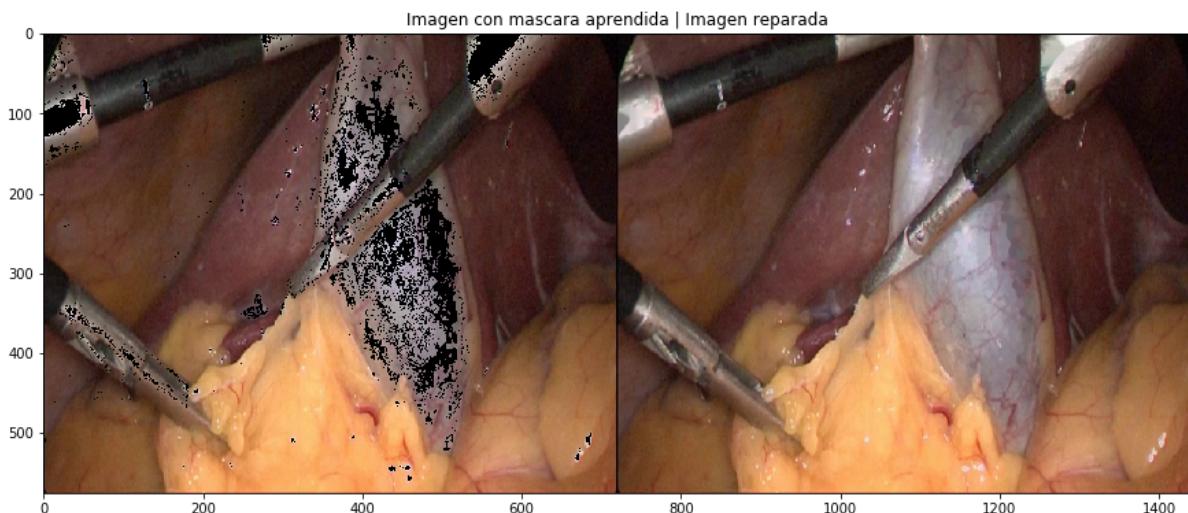
Con estos resultados, decidí tomar el clasificador y crear la máscara correspondiente a la imagen. Notar que para este resultado, el 80% de los píxeles fueron parte del entrenamiento.



La máscara aprendida, pese a que el 80% de los píxeles estuvieron presentes en el entrenamiento, no es de buena calidad. Al realizar el inpainting, se llega a lo siguiente:



Poniendo a prueba el nuevo enfoque para la imagen del otro video, se llega a lo siguiente:



Para terminar, veamos a continuación la imagen original y las dos reconstrucciones según el enfoque de umbralizar (arriba) vs aprender (abajo):

Imagen original | Imagen sin especlaridades

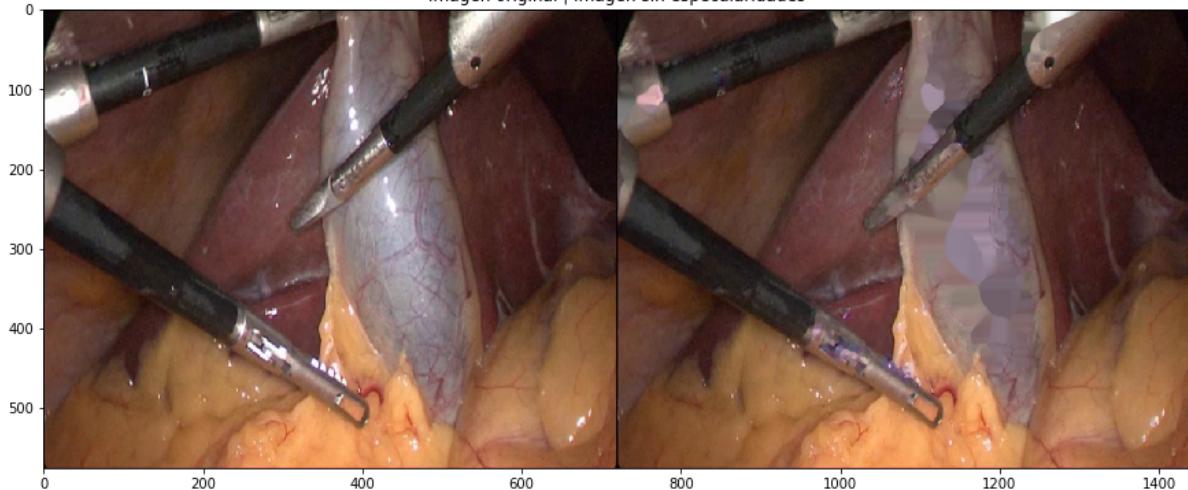


Imagen original | Imagen sin especlaridades

