جامعة دمشق

كلية هندسة المعلوماتية

السنة الخامسة

اختصاص الذكاء الصنعي

21/12/2023

تقرير مشروع الرؤية الحاسوبية

اشراف:

المهندسة نور الحكيم

اعداد:

عبد الله محمد عبد الناصر الزبداني

محمد عثمان بسام دياربكرلي

طوني ابراهيم بطرس

عبد الرؤوف كمال حسحس

Contents

مقدمة

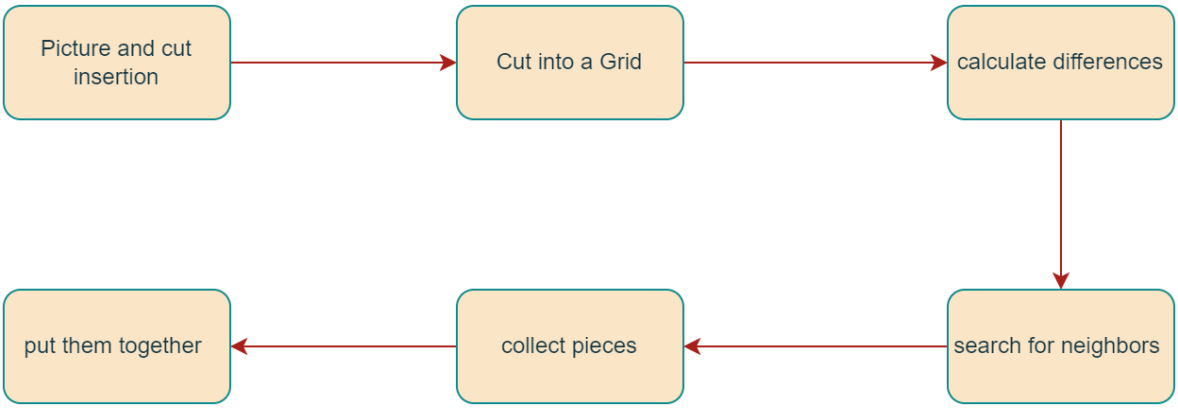
في هذا التقرير سيتم شرح كيف وصلنا لحل كل من الgrid puzzle والjigsaw puzzle باستخدام ال computer vision اعتمادا على توابع مكتبة OpenCV. والخوارزميات المتبعة والحلول التي توصلنا لها.

الطلب الأول:

Grid puzzle without hint

طريقة الحل:

ال pipeline المتبع



Picture and cut insertion:

يقوم المستخدم بإدخال كل من الصورة المطلوبة (الصورة غير مقطعة) مع عدد التقطيع عاموديا وعدد التقطيع افقيا، تمت الاستعانة بالتابع read_imege.

Cut into a grid:

تقطع الصورة المدخلة ال قطع عددها يساوي عدد القطع المدخل، واثناء عملية التقطيع يقوم تابع التقطيع get_pieces بانشاء objects خاصة بكلاس piece الذي يخزن مجموعة من المعلومات التي نحنا بحاجتها حتى تكون خوارزميتنا فعالة، المعلومات المخزنة:

pieceNum: رقم يحدد القطعة.

size_vertical و size_horizontal: أبعاد القطعة.

pieceChn: عدد القنوات في القطعة.

pieceStart: نقطة البداية للقطعة.

pieceData: مصفوفة NumPy تحتوي على بيانات البكسل الخاصة بالقطعة.

pieceTotal: العدد الإجمالي للقطع.

SideUp، وsideRight، وsideDown، وsideLeft: تخزين بيانات البكسلز للجوانب الأربعة للقطعة.

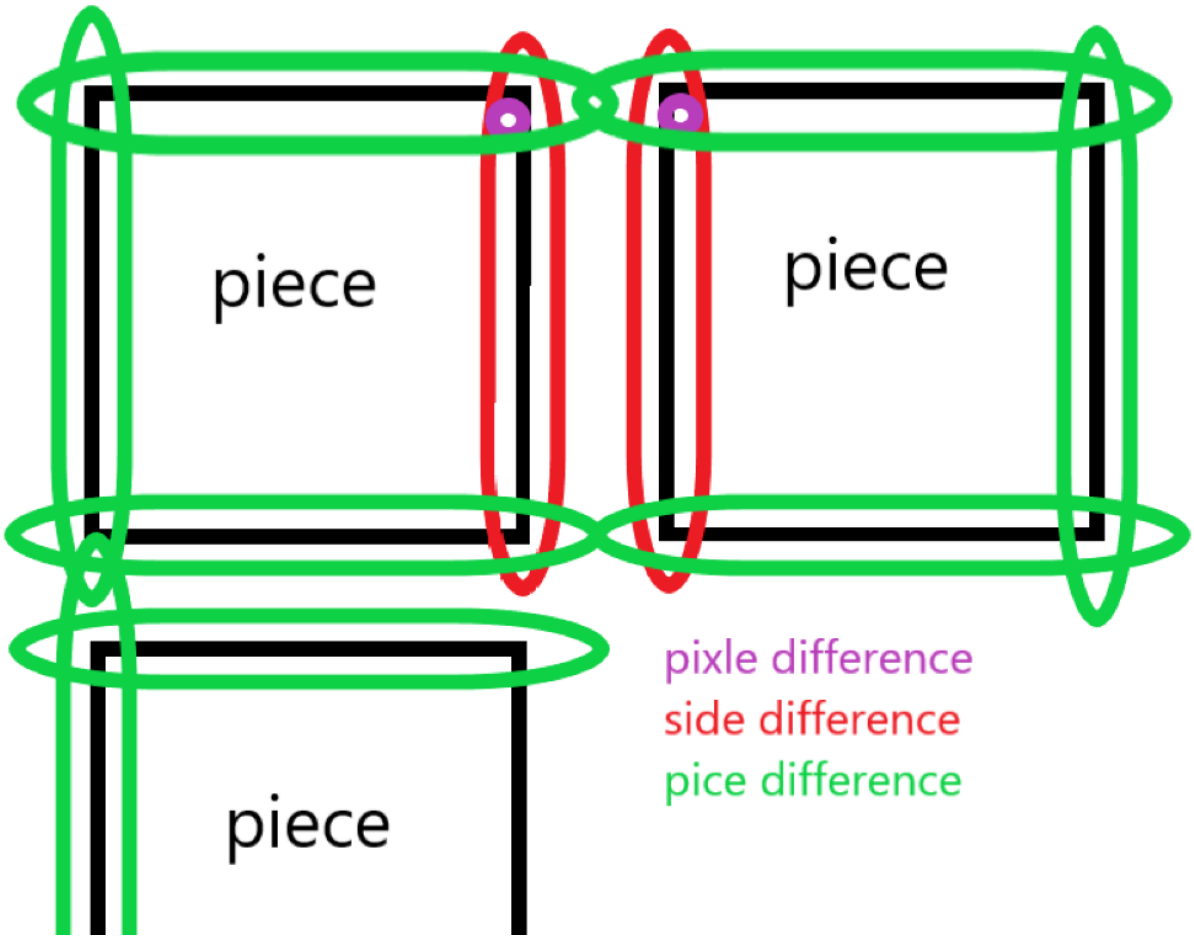
sides: قائمة تحتوي على مراجع لجوانب القطعة الأربعة.

difference: قائمة لتخزين الاختلافات بين هذه القطعة والقطع الأخرى.

neighbors: قائمة لتخزين مؤشرات القطع المجاورة في اتجاهات مختلفة.

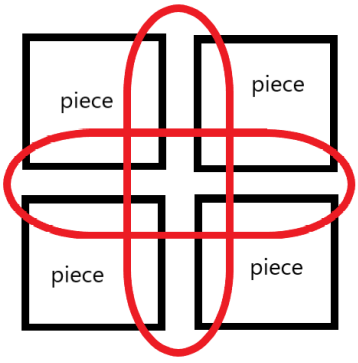
Calculate differences:

بالاعتماد على التابع piece_difference نقوم بحساب الاختلاب بين القطع عن طريق الفرق بين البكسلات بالحواف عند الجوان الأربعة، حيث التابع السابق أيضا يعتمد على التابع side_difference يقوم بحساب الفرق بين طرفين وذلك أيضا بالاعتماد على التابع pixel_difference الذي يقوم بحساب الفرق بين بكسلين.



Search for neighbors:

يقوم التابع find_neighbors بإيجاد القطع المتقاربة من بعضها ذلك من خلال تجميع القطع التي أطرافها تملك اقل اختلاف فيما بينها.



side difference < 0.6

Collect pieces:

مرحلة وضع القطع مع بعضها من اجل تجميعها بصورة كاملة.

Put them together:

تجميع جميع القطع بصورة واحدة من اجل عرض النتيجة النهائية.

Grid with hint

طرق الحل:

الطريقة الأولى:

تعتمد على استخراج الخصائص من الصور المكونة للشبكة وللصورة الhint ثم مطابقة هذه الخصائص بين الصورة الhint والصور المكونة للشبكة من ثم تشكيل صورة الخرج بناءا على نتائج المطابقة بأفضل طريقة ممكنة

حيث تم الاستعانة ب class FeaturesExtractor لاستخراج الخصائص، يوفر هذا الصف عدة طرق لاستخراج الخصائص مثل

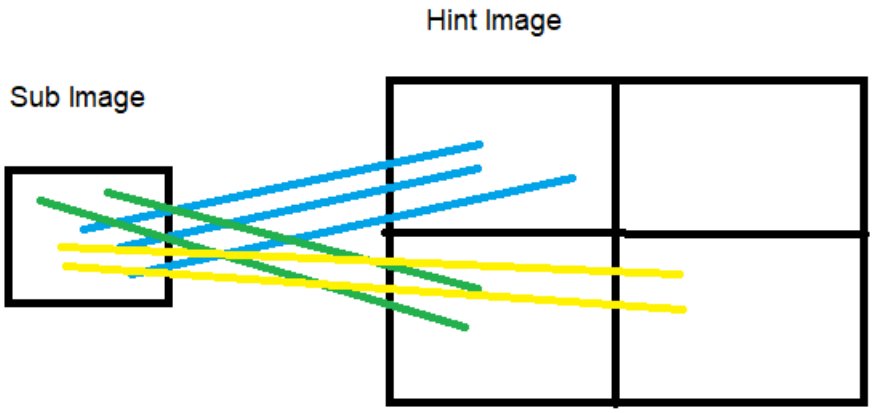
ORB,SURF ,SIFT

و تمت الاستعانة ب class Matcher من اجل اجراء عملية المطابقة بين الخصائص، يوفر هذا الصف الطرق التالية من اجل اجراء المطابقة

KNN , BF_MATCHER , FLANN , BRUTE_FORCE

خوارزمية الحل:

1- نقوم بقراءة الصورة ال hint وتقطيعها وتخزين نتائج التقطيع
2- نقوم بانشاء مصفوفة أخرى بناءا على نتائج التقطيع والتي تعبر عن shuffled grid
3- نقوم باستخراج الخصائص للصورة الhint وللل shuffled grid
4- نقوم بعمل مطابقة للخصائص بين الصورة ال hint وقطعة من الشبكة
5- نقوم بانشاء مصفوفة تعبر عن اعداد الخصائص التي تمت مطابقتها بين القطعة sub image مع كل قطعة من الصورة الhint



Matches=[(0,3),(1,0),(2,2),(3,2)]

6- نقوم بترتيب المصفوفة matches بناءا على عدد المطابقات
7- نكرر الخطوات 4,5,6 بالنسبة لجميع قطع الshuffled grid
8- نقوم بإنشاء صورة الخرج اعتمادا على تابع dfs حيث في كل خطوة يقوم بمحاولة إضافة أكثر صورة لها مطابقات بالمكان المناسب، بعد الانتهاء من محاولة التركيب (عند وصول عدد القطع المركبة الى عدد قطع الgrid الاصلية) نقوم بحساب دقة الحل، حيث إذا كانت الدقة غير مناسبة يقوم تابع الdfs بتركيب صور أخرى الى ان نصل الى الدقة المرجوة
9- نقوم بحساب الدقة بناءا على القيم اللونية لصورة الخرج والصورة الأساسية

الطريقة الثانية:
خوارزمية الحل:

1- نقوم بقراءة الصورة ال hint وتقطيعها وتخزين نتائج التقطيع
2- نقوم بإنشاء مصفوفة أخرى بناءا على نتائج التقطيع والتي تعبر عن shuffled grid
3- نقوم بحساب التشابه بين قطعة من shuffled grid وكل قطع ال hint
4- نقوم بإيجاد أكبر تشابه ممكن وتخزين المعلومات اللازمة عنه
5- نكرر الخطوات 3,4 على كل قطع ال shuffled grid
6- نقوم بانشاء صورة الخرج بناءا على المعلومات التي تم استخراجها اثناء تنفيذ الخطوات السابقة

المقارنة:

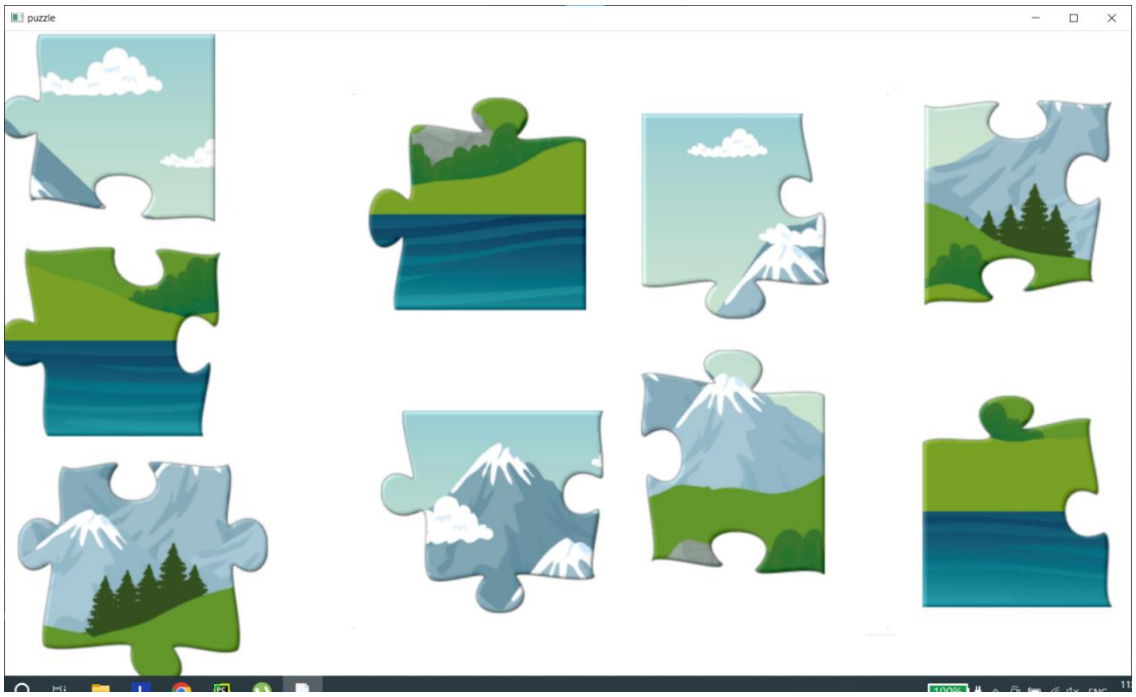
	Without hint	With hint
محاسن	مع الصور الملونة يسهل على الخوارزمية التفريق بين القطع بغض النظر عن حجم التقطيع من الممكن ان نصل لحجم تقطيع 10*10 والخوارزمية تعطي نتائج صحيحة بنسبة 90%، بحجم تقطيع اقل مثلا 6*6 يكون دقة الحل 100%.	الحل باستخدام مقارنة الألوان بغض النظر عن الصورة يعطي حل دقيق 100%.
مساوئ	دقة الحل تعتمد على الصورة المدخلة وحجم التقطيع، حيث الصور التي الألوان فيها متشابهة تصعب على الخوارزمية التفريق بين القطع وذلك يادي الى ترتيب صحيح، ولكن بالتلاعب بحجم التقطيع من الممكن ان نصل لحل شبه صحيح.	الحل باستخدام الFeature extraction يعطي حل سيء لان القطعة تعطي match على عدة مناطق من الصورة hint مما يادي ان الخوارزمية لا تستطيع وضع الصور بالمكان المناسب.

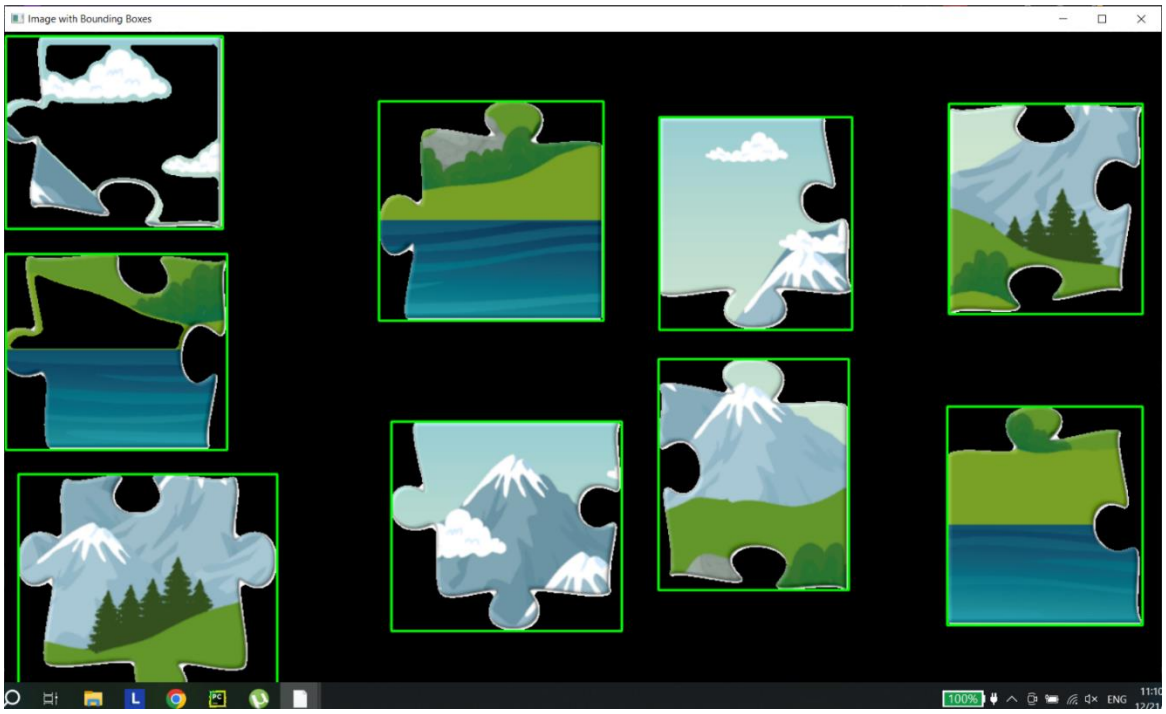
الطلب الثاني:

Jigsaw Puzzle

نقوم بإجراء الخطوات التالية كمعالجة أولية للصور التي تحتوي على قطع ال puzzle حيث سوف تطبق الخطوات التالية في كلا الطلبين

1- نقوم بقراءة الصورة
2- نقوم بإنشاء غطاء ثنائي باستخدام adaptiveThreshold وذلك من اجل عزل قطع الpuzzle عن خلفية الصورة
حيث نقوم بتحويل الصورة الى gray scale ثم تطبيق adaptiveThreshold ثم تطبيق فلتر opening من اجل التخلص من ال noise بعد ذلك نقوم بتطبيق فلاتر dilate وclosing وerosion
3- نقوم باكتشاف محيطات قطع الpuzzle وترتيبها بالنسبة للأكبر ثم تطبيق median filter من اجل تنعيم قطع الpuzzle
4- هنا يكون القناع جاهز من اجل عملية استخراج كل قطعة على حدي





القسم الأول:

بعد استخراج كل قطعة على حدى نقوم بإيجاد نوع القطع حيث قد تكون القطع (قطعة زاوية، قطعة طرف، قطعة داخلية)

خوارزمية إيجاد نوع القطعة:

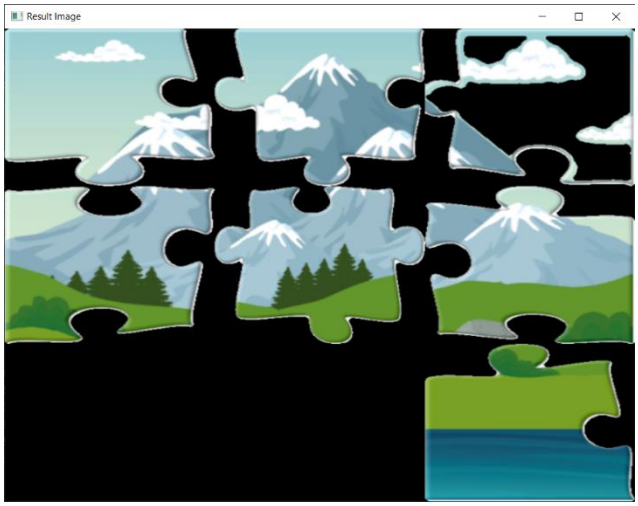
1- نقوم بحساب distance transform
2- نقوم بحساب centroid
3- نقوم بإيجاد حواف القطعة باستخدام harris corner
4- نقوم بإيجاد normal vector كل نقطة من الharris corner والcentroid ثم نقوم بتخزينها في مصفوفة
5- نقوم بإيجاد العتبات في المصفوفة السابقة
6- نقوم بحساب الزوايا ثم نختار أربع زوايا التي تعبر عن الزوايا الفعلية لقطعة ال puzzle
7- بناءا على المعلومات السابقة يمكننا معرفة شكل قطعة ال puzzle

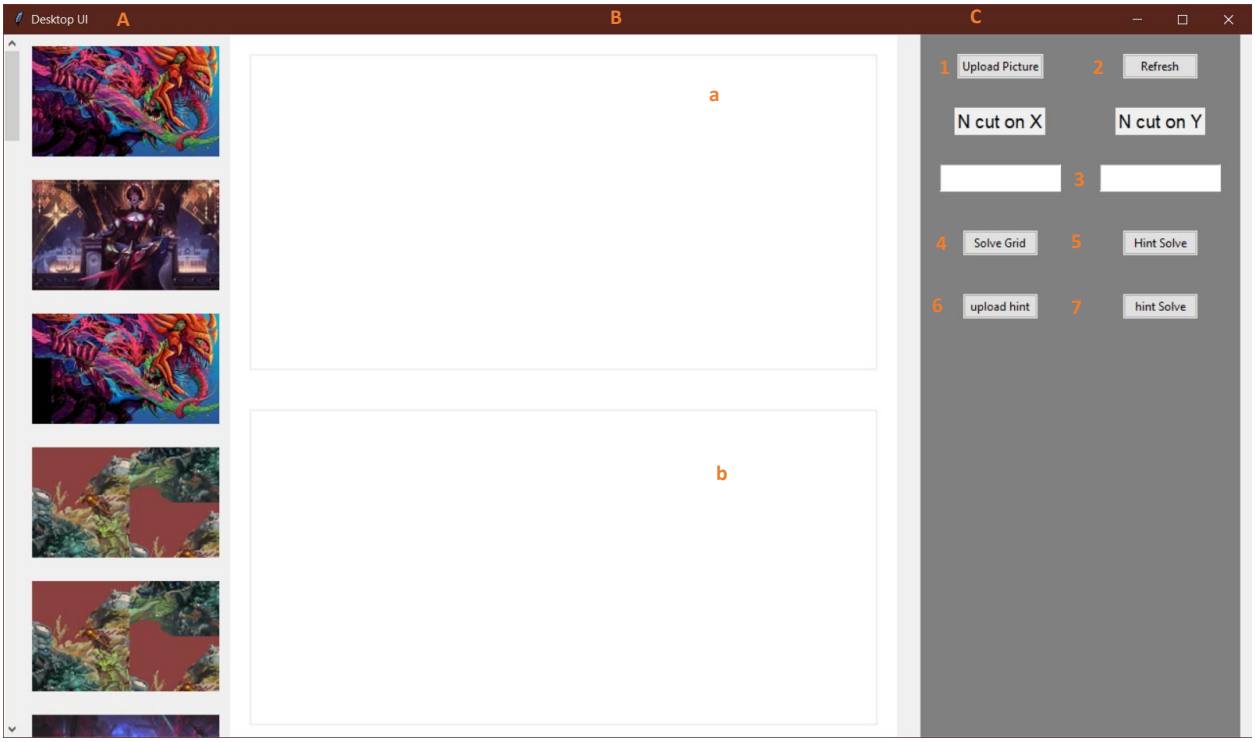
ثم نقوم بتركيب الصورة على أساس اشكال والوان القطع

القسم الثاني:

خوارزمية الحل:

1- نقوم بقراءة الصورة ال hint وتقطيعها وتخزين نتائج التقطيع
2- نقوم بإيجاد قطع ال puzzle
3- نقوم بحساب التشابه بين قطعة من قطع الpuzzle وكل قطع ال hint
4- نقوم بإيجاد أكبر تشابه ممكن وتخزين المعلومات اللازمة عنه
5- نكرر الخطوات 3,4 على كل قطع ال puzzle
6- نقوم بإنشاء صورة الخرج بناءا على المعلومات التي تم استخراجها اثناء تنفيذ الخطوات السابقة





A. قائمة النتائج:

تحتوي على الخرج النهائي بعد تطبيق أحد الخوارزميات للحل، والذي هو الصور الناتجة بعد الحل.

B. قائمة العرض:

a. مربع عرض 1:

عرض الصورة المختارة من قبل المستخدم.

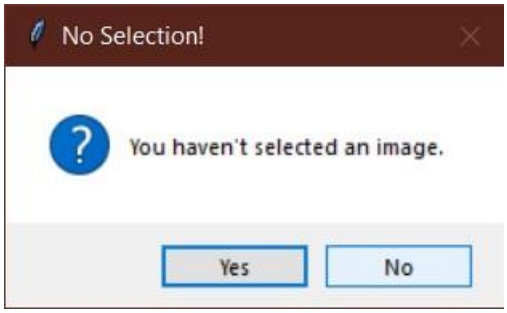
b. مربع عرض 2:

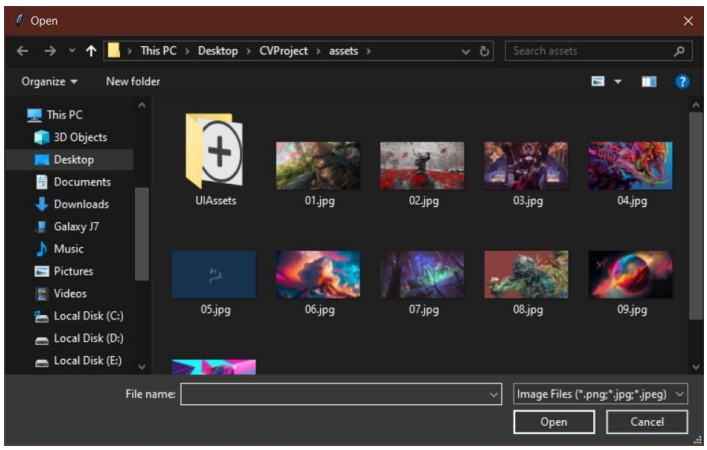
عرض الصورة الناتجة بعد تطبيق أحد الخوارزميات.

C. قائمة التحكم:

1. Upload picture:

لاختيار وتحميل صورة من ملفات المستخدم.





2. Refresh:
إعادة تحميل النتائج في قائمة النتائج.

3. X and Y cut:
ادخال عدد القص عاموديا وافقيا.

4. Solve grid:
حل الصورة المدخلة بدون HINT على طريقة الgrid puzzle.

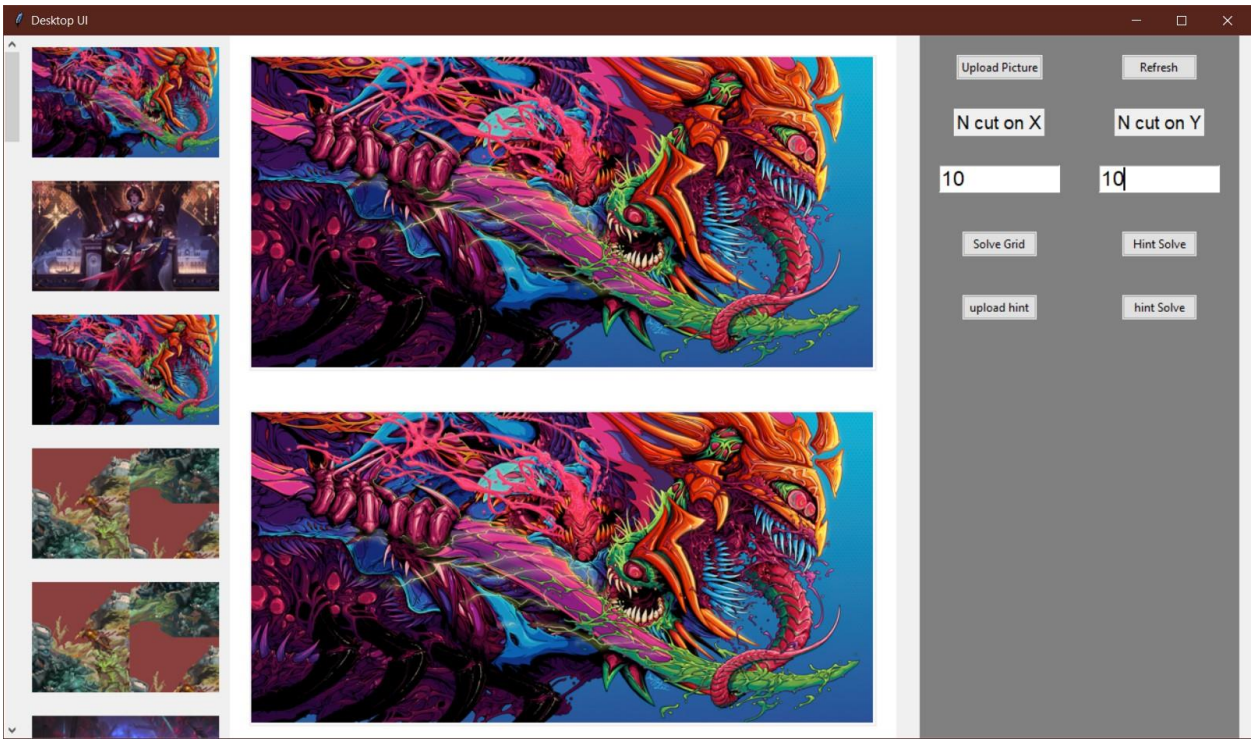


Image1 full colored picture gives greater results

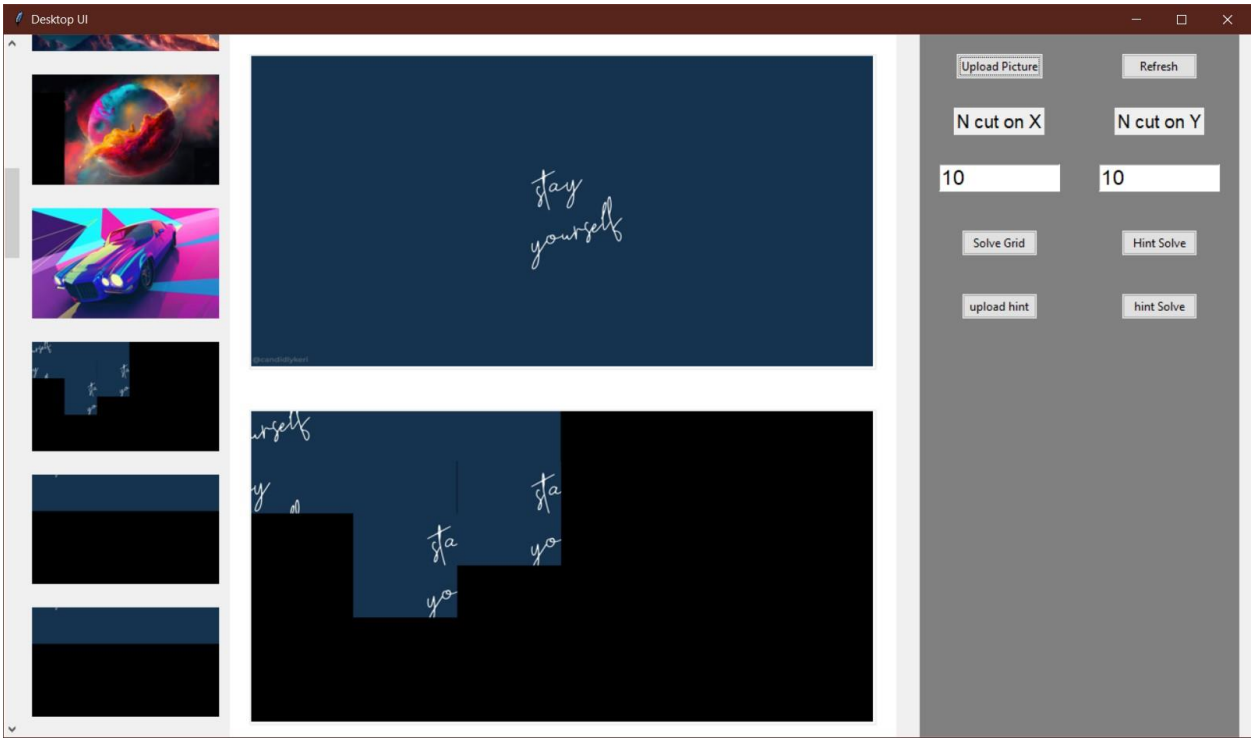
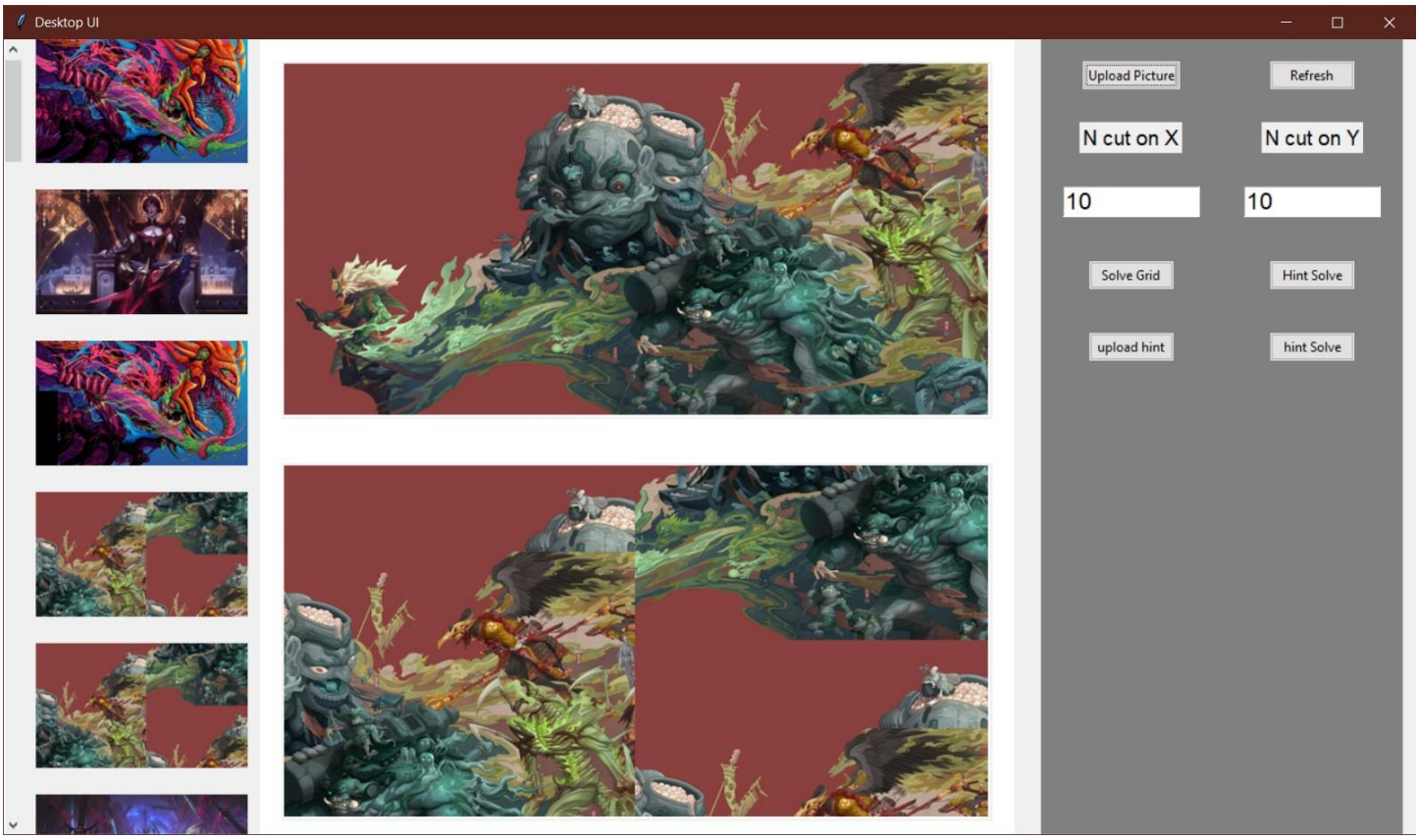


Image 2 one few colors image harder to solve not a good result



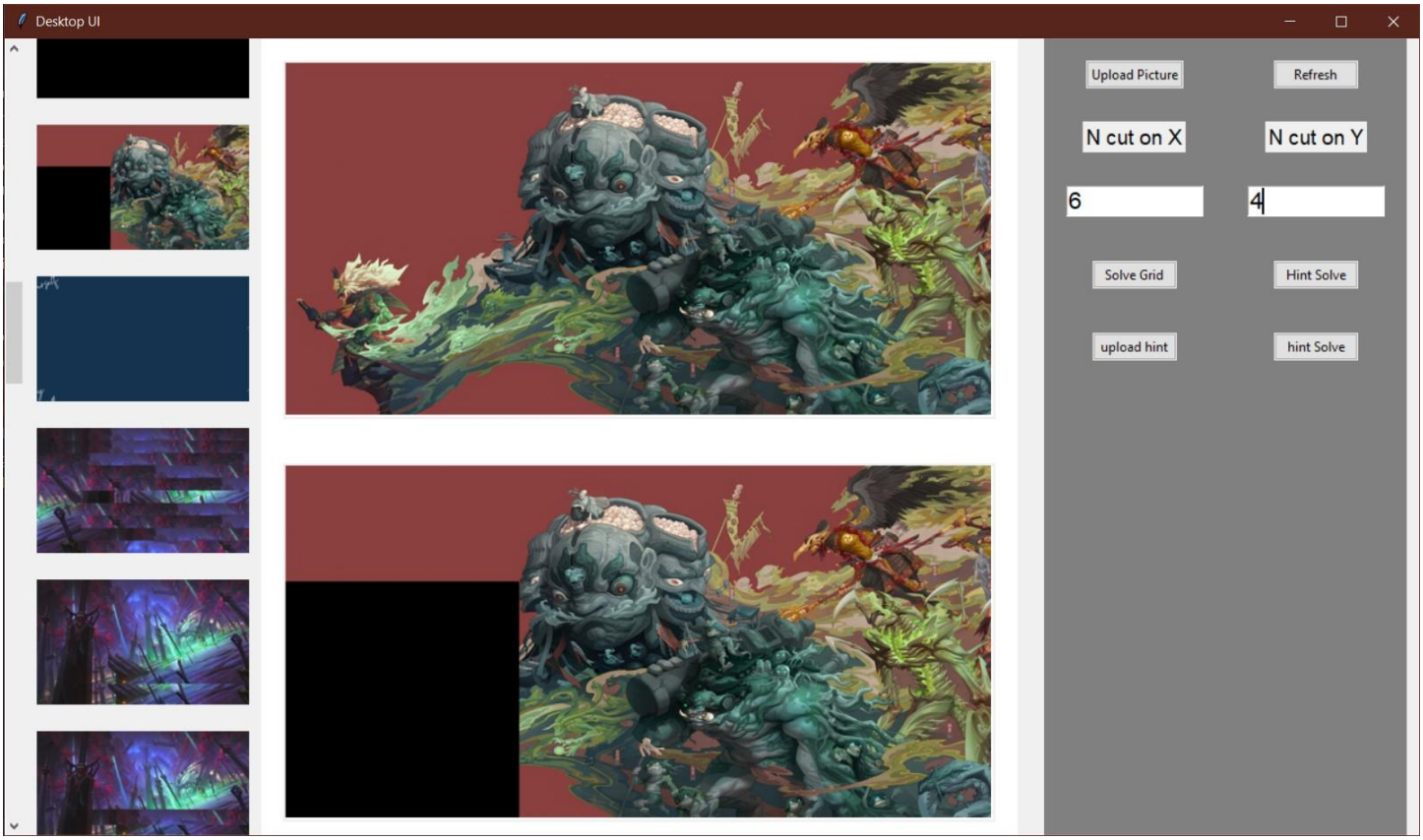


Image 1 different cutting might give a better results

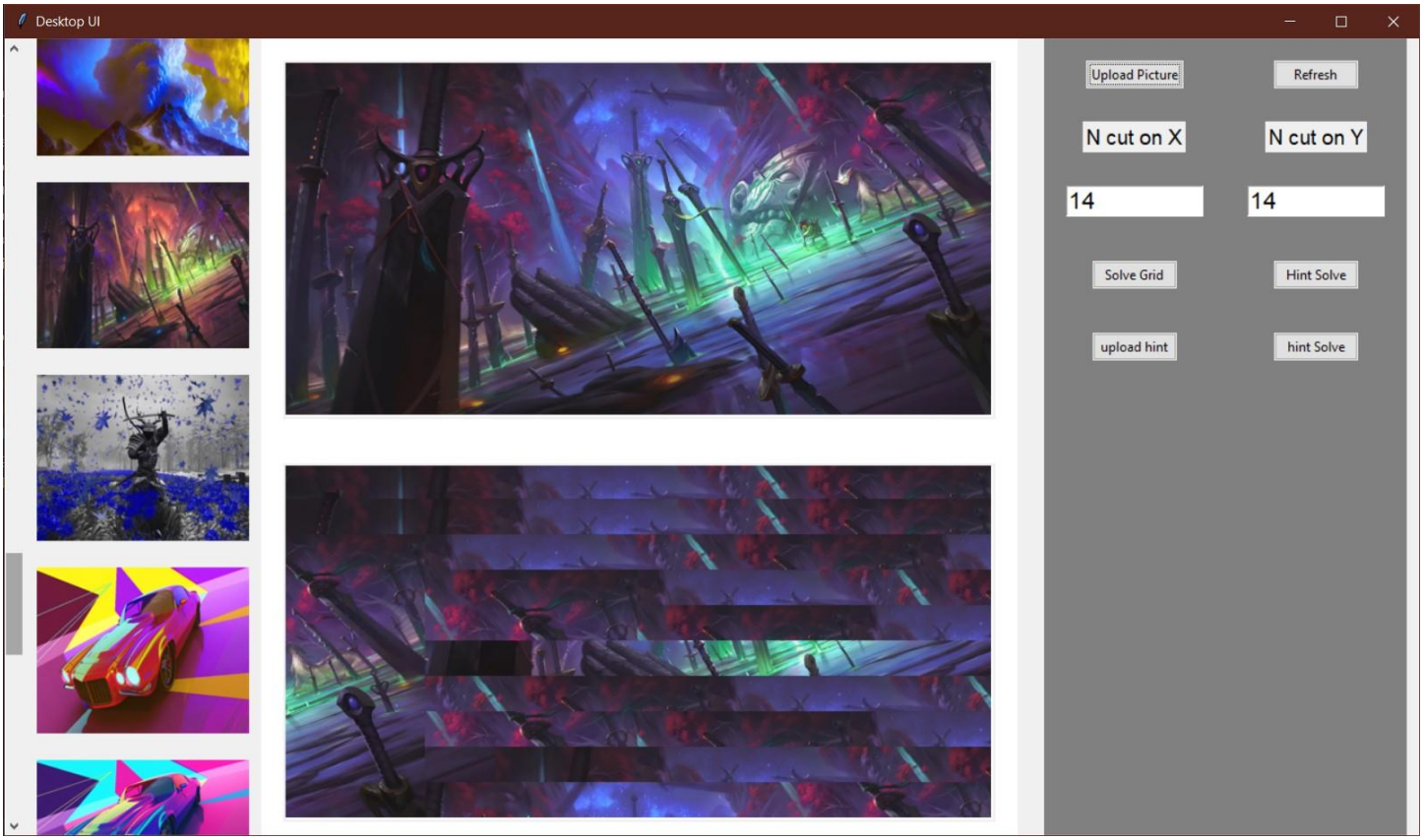
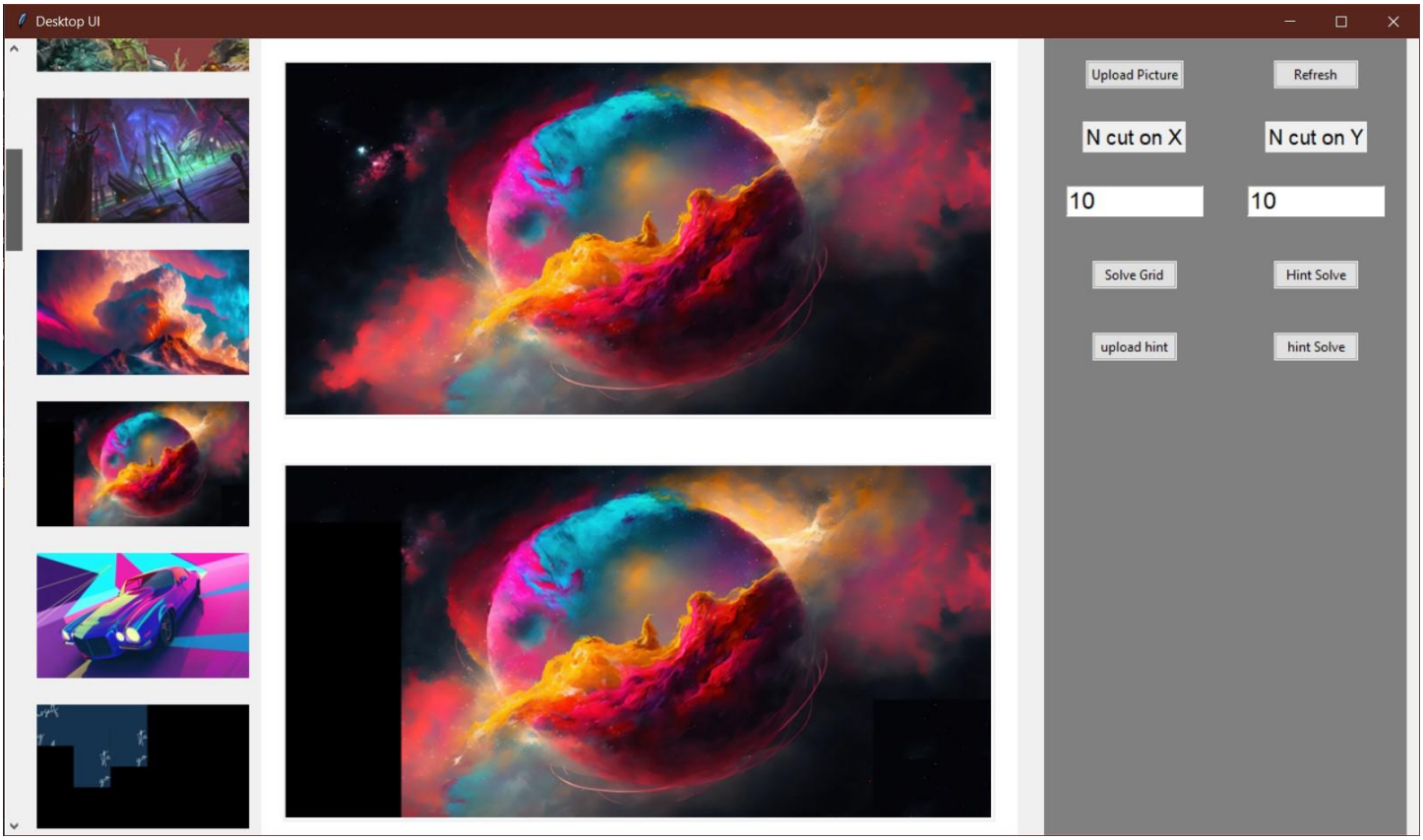
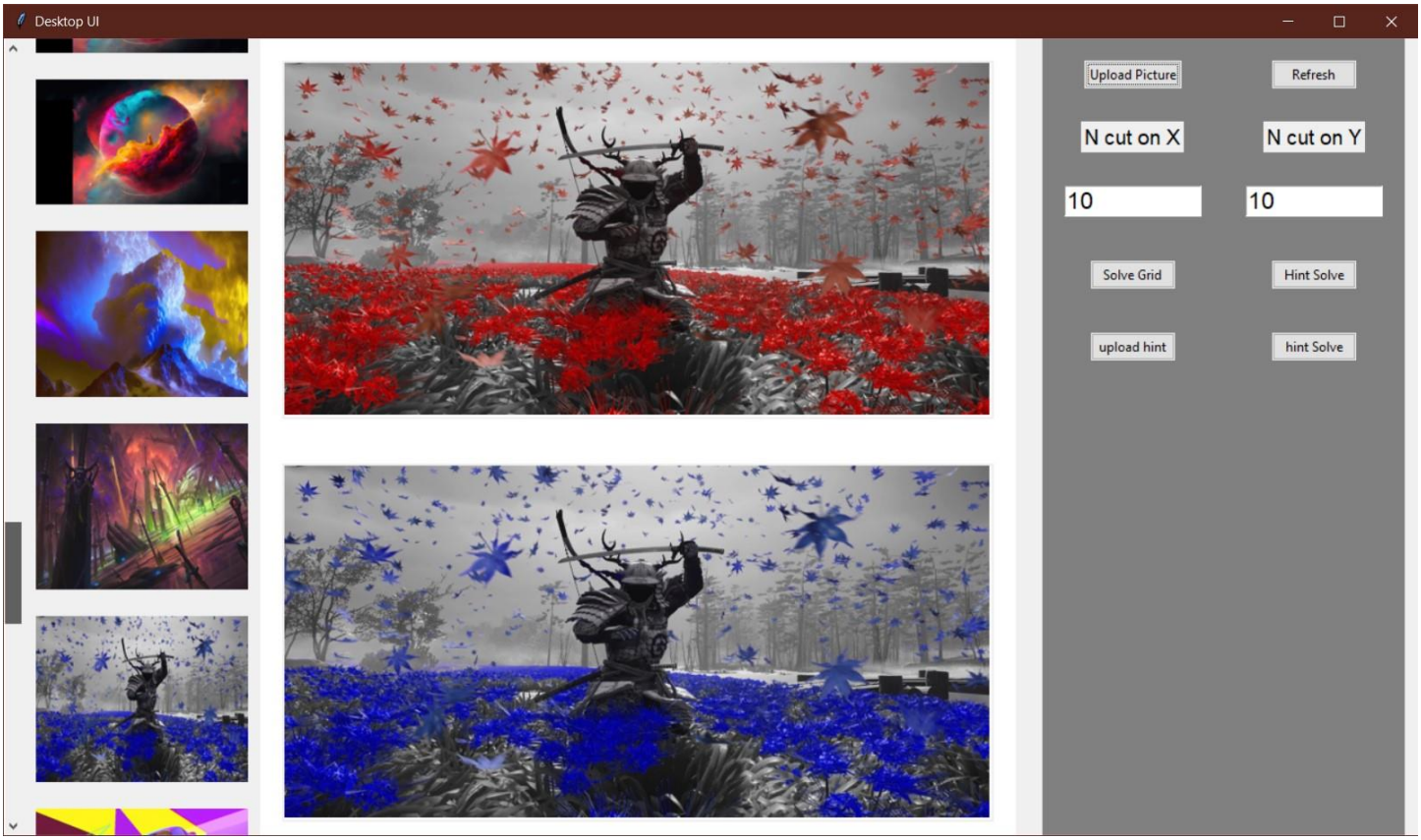


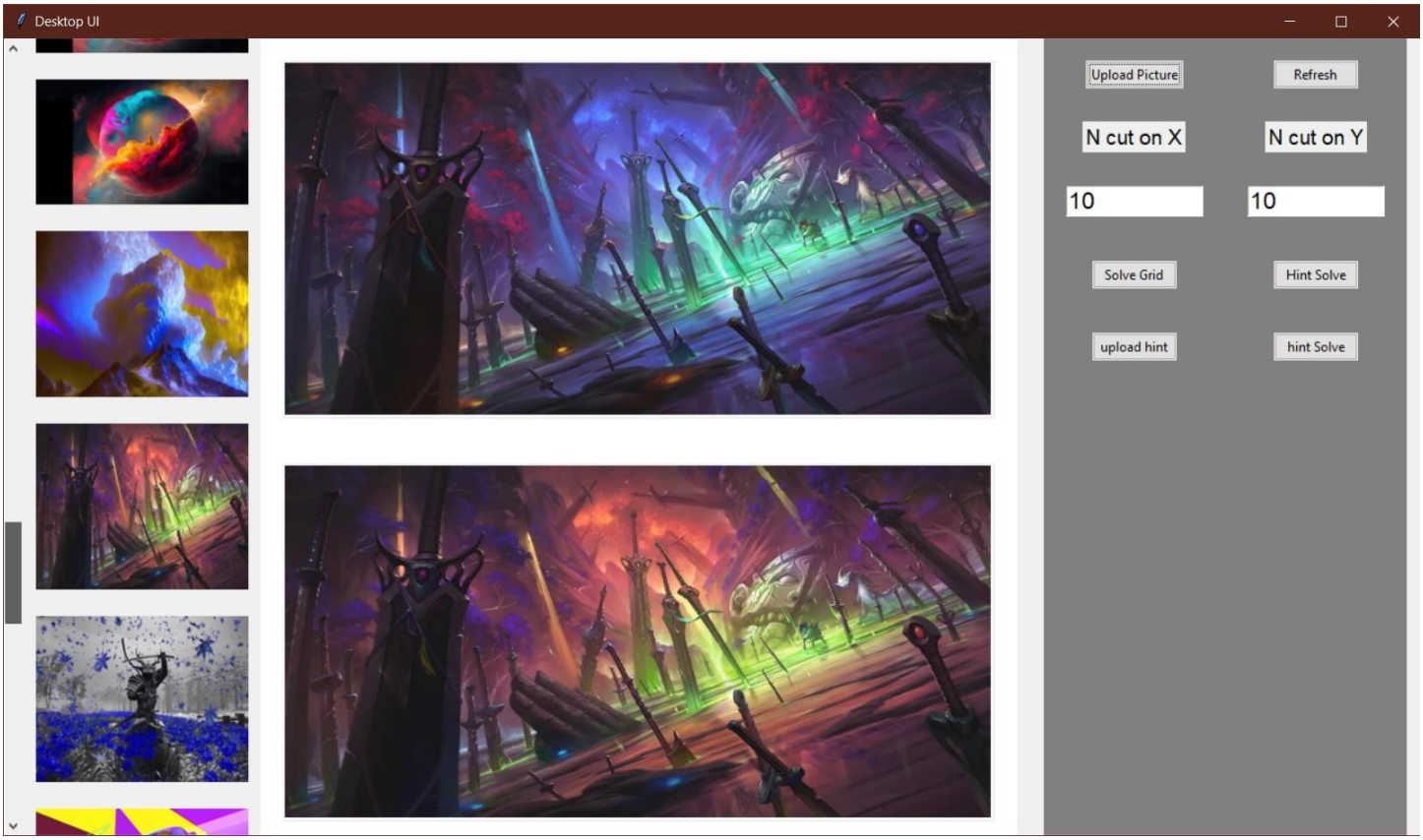
Image2 more cut also gives bad results

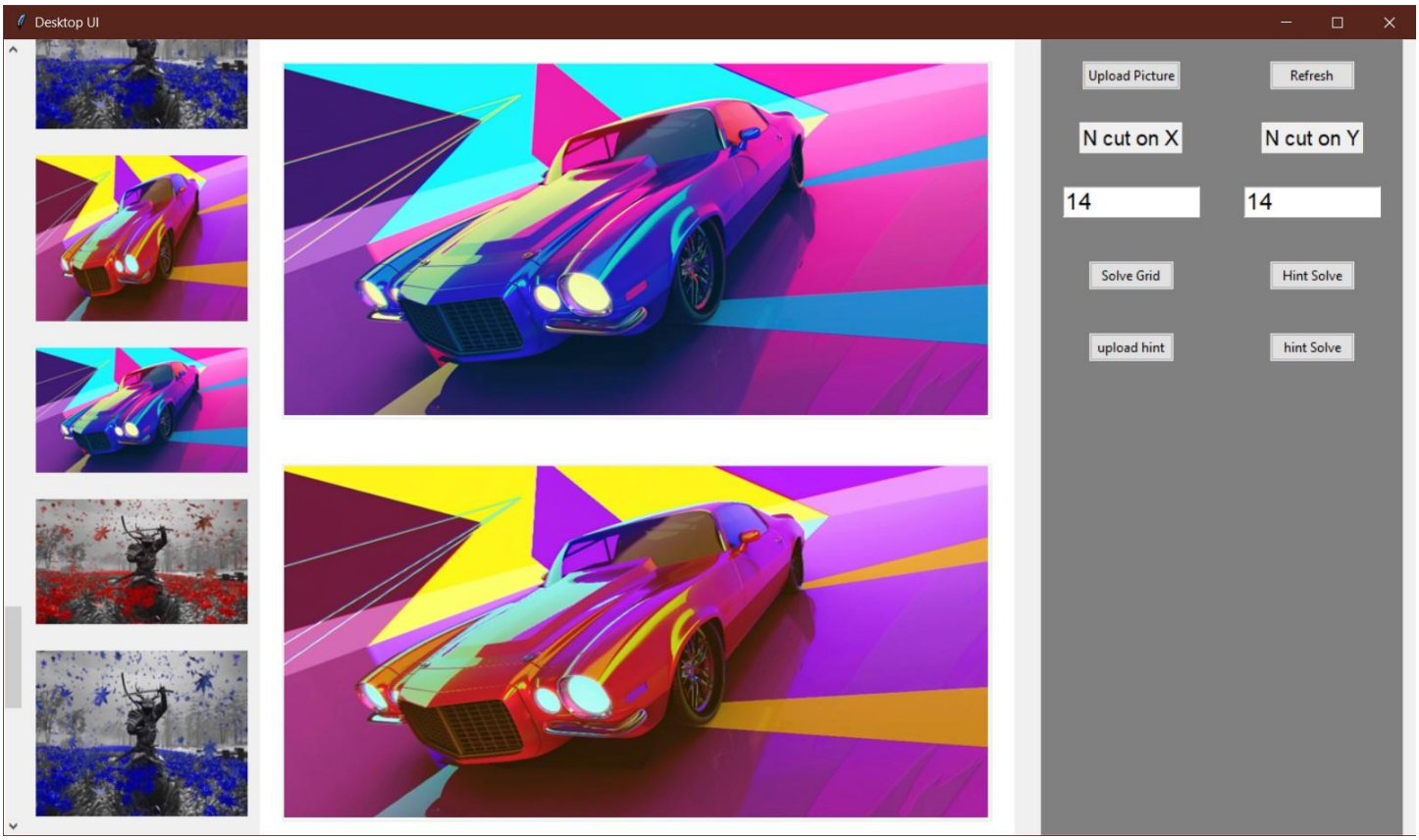


5. Hint solve:

حل الصورة المدخلة مع HINT على طريقة ال grid puzzle.







6. Upload hint:

تحميل صورة HINT من ملفات المستخدم من اجل حل الصورة المدخلة ك jigsaw puzzle.

7. Hint solve:

حل الصورة المدخلة مع HINT على طريقة ال jigsaw puzzle.