**Обработка на изображения**

**І. Image Composition**

*1. ImageCompose- първата функция е ImageCompose. Чрез нея се наслагват две или повече изображения. Второто изображение е в списък. Задаваме неговата степен на прозрачност. Колкото по-малко е числото, толкова по-прозрачно е изображението. Добавяме изображение от Insert/Picture -> From file*

**ImageCompose [ *compose1, { compose2*, .3 } ]**

*2. Blend – следващата функция – Bland смесва изображенията. С помощта на функцията Manipulate се постига бавно преливане между изображенията.*

**Manipulate[ Blend[ { *tree1, tree2, tree3, tree4* }, a], {a, 0, 1.2}]**

*3. ImageCollage – следващата функция създава колаж от списък изображения. С опцията ImagePadding задаваме рамка на изображенията. Цифрата показва дебелината на рамката. С опцията Padding задаваме цвят на рамката, а с Method избираме начина на подредба на изображенията.*

**ImageCollage [ { *collage1, collage2, collage3, collage4, collage5* }, ImagePadding -> 15, Padding -> Orange, Method -> "Columns" ]**

**ІІ. WordCloud – функция, която запълва с думи определена форма (шаблон).**

В случая ще запълним шаблон с имената на държавите. С таговете “Name” и “Population” се задават имената.

**data = EntityValue [ CountryData [], { "Name", "Population" } ];  
WordCloud [ data, *word* ]**

**ІІІ. Face Detection**

Функцията FindFaces намира и маркира лица от дадено изображение (чиято приблизителна пикселна ширина е в определени граници. В нашия случай е от 50 до безкрайност.). Задаваме две променливи – *i* и *boxes*. Знак за безкрайност се записва с Esc-inf-Esc. Функцията *Show* показва графики с определени опции, а с Graphics задаваме начина, по който се отбелязват лицата – с червен квадрат.

**i = *face*;  
boxes = FindFaces [ i, {50,∞ } ];  
Show [ i, Graphics [ { EdgeForm [ {Red, Thick } ], Opacity [0], Rectangle @@@ boxes } ] ]**

**ІV. Motion Blur Effect – Създава ефект на размазване при движение**Имаме три променливи – *i*, *blurred* и *mask*. На променливата *i* присвояваме изображението, на *blurred* – ефектът върху изображението, а на *mask* – маската на изображението.  
- ImageEffect прилага определени ефекти към дадено изображение. В нашия случай – към изображението, което е присвоено на променливата i.  
- DeleteSmallComponents – замества малки свързани компоненти с фонови пиксели  
- Binarize – намира пикселите, в които преобладава червеният цвят  
- ImageCompose – функция за наслагване на изображения  
- SetAlphaChannel – добавя към изображението (замъглената) маска  
- Blur – замъглява изображението

**i = *motion*;  
blurred = ImageEffect [ i, { "MotionBlur", 30, 0 Degree } ];** //прилагаме замъглен ефект върху изображението **mask = Dilation [ DeleteSmallComponents @ Binarize [ i, #[[1]] > .65 ( # [[2]] + # [[3]] ) & ], 25 ];** //задаваме маска на изображението, като за основен цвят определяме червения **ImageCompose [ i, SetAlphaChannel [ blurred, Blur [ mask, 50 ] ] ]** //с функцията за наслагване на изображения добавяме (замъглената) маска

**V. Simplify an Image Mask – последният ефект, който ще направим, е да извлечем определена част от дадено изображение; създава се маска на изображение**

Имаме шест променливи – *i, b, skeleton, pruned, mask* и *res*. На променливата *i* присвояваме изображението, на *b* – основният цвят - червен, на *skeleton* – схемата на изображението, на *pruned* – изрязяният обект, на *mask* – възстановената форма на обекта и на *res* –маската на изображението.

- DeleteSmallComponents – замества малки свързани компоненти с фонови пиксели  
- FillingTransform -   
- ChanVeseBinarize – определяне на основния цвят  
- SkeletonTransform - Дава схемата на трансформиране на изображението  
- Pruning – премахва отдалечените разклонения на обекта, като задава стойностите им в черно  
- InverseDistanceTransform – възстановява формата на обекта, използвайки схемата му  
- SetAlphaChannel – добавя към изображението маска  
- Grid – подрежда обектите в двойна решетка

**i = *mask*;  
b = DeleteSmallComponents @ FillingTransform @ ChanVeseBinarize [i,"TargetColor" -> Red];** //определяме основния цвят **skeleton = SkeletonTransform [ b ];** //задаваме схема на изображението **pruned = Pruning [ skeleton, 1, 5 ];** //премахва разклоненията на обекта, като задава стойностите им в черно **mask = InverseDistanceTransform [ pruned ];** // възстановява формата на обекта, използвайки схемата му **res = SetAlphaChannel[i, mask];** // добавя маска към изображението **Grid[{{i, b, skeleton}, {pruned, mask, res}}]** // подрежда обектите в двойна решетка