+ - реализовано

-+ - почти реализовано

1. +rotate(degree = 90);  
   degree – угл наклона. Может быть положительным и отрицательным.
2. +crop(width = null, height = null, startX = null, startY = null, isProp = false);  
   startX, startY – начало координат откуда начинать обрезать.   
   width – ширина исходного изображения  
   height – высота исходного изображения  
   isProp – булевая переменная, если true, то проверяет возможно ли просто уменьшить пропорционально изображение. Если да, то вызываеться метод resize(width, height);
3. frame(color, width, isRounded = false);  
   color – цвет рамки  
   width – ширина рамки  
   isRounded – булевая переменная, которая указывает будет ли рамка закругленная. По умолчанию false.
4. +insert(image, position, offset, offsetY); - наложение изображения  
   image – изображение(как объект, на котором тоже можно использовать методы его класса)  
   position – позиция где должно находиться второе изображение. Принимает слова такие как – ‘top-left’ , ‘top’ ‘top-right, ‘left, ‘center, ‘right’ ‘bottom-left’ ‘bottom’ ‘bottom-right’
5. Методы фильтров пока не знаю. Какой нибудь отдельный класс который унаследует класс картинки и в котором реализованы фильтры. Но это не точно надо ещё подумать.  
   +gamma(value) - Выполняет операцию гамма-коррекции на текущем изображении.  
   +bright(value) – изменяет яркость изображения  
   +contrast(value) – изменяет контраст изображения  
   +colorize(red, green, blue) - Изменение значения цвета RGB текущего изображения по заданным каналам **красного** , **зеленого** и **синего**   
   +makeGrayscale();  
   нечего не принимает, просто делает картинку черно-белой.  
   +invert() (фильтр) - Меняет все цвета текущего изображения.

1. +mirror(value = ‘h’);  
   метод отзеркаливает изображение.  
   value – принимает два параметра ‘h’ и ‘v’(горизонтально, вертикально соответственно)
2. Оптимизация. Тоже самое что и шестой пункт.
3. +opacity(value);  
   value – принимает значение от 0 до 100(0 – полностью невидим. 100 – полностью видим).
4. makeGIF(imgArr, param1, param2);  
   imgArr – массив изображений которые мы будем менять с помощью param1, param2, точно ещё не знаю надо подумать.
5. +resize(width, height);  
   тут всё ясно. Возможно если получиться то можно сделать увеличение картинки с минимальной потерей качества.
6. border(c1, c2, c3, c4, c5); - находит границы изображения  
   с1,с2,с3,с4,с5 – коэффициенты
7. histogram(); - возвращает график в виде изображения со всеми цветами.
8. +blur(value = 1);
9. +filters(filters) – принимает строку типа ‘opacity- contrast’ и применяет эти стили.
10. +outPut() – выводит изображение
11. +Save() – сохранить изображение в папку