+ - 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(type = ‘type-1’);  
   type – перечень рамок на выбор(‘type-1’, ‘type-2’, ‘type-3’…)
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. +insertMerge($img, $opacityValue, $pos = 'top-left', $offsetX = 0, $offsetY = 0) – налаживает изображение с прозрачностью
6. +insertGrayscale($img, $pos = 'top-left', $offsetX = 0, $offsetY = 0) – налаживает изображение и делает его черно-белым
7. +insertResize($img, array $size, $pos = 'top-left', $offsetX = 0, $offsetY = 0) – налаживает изображение и изменяет его размер
8. +insertCrop($img, array $values, $pos = 'top-left', $offsetX = 0, $offsetY = 0) – налаживает изображение и обрезает его
9. +text($text, $startX = 0, $startY = 0) – налаживает текст на изображение
10. +pixelate(size = 1) – устанавливает размер пикселей
11. 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 = 40) - Выполняет операцию гамма-коррекции на текущем изображении.  
    +bright(value = 40) – изменяет яркость изображения  
    +contrast(value = 50) – изменяет контраст изображения  
    +colorize(red = 40, green = 40, blue = 40) - 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 = 60);  
   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, heigh = nullt);  
   тут всё ясно. Возможно если получиться то можно сделать увеличение картинки с минимальной потерей качества.
6. +-borders($type = 'color', $format = 'jpg', $quality = 90, $sigma = 1, $sobelK = 1, $low = 50, $high = 150, $size = 2); - находит границы изображения
7. +histogramEqualization(type = ‘grayscale’);  
   можно сказать что это очень модный фильтр  
   type – принимает значения ‘color’, ‘grayscale’, ‘color-to-grayscale’  
   ‘color’ – пользователь указывает что изображение цветное  
   ‘grayscale’ – пользователь указывает что изображение черно-белое  
   ‘color-to-grayscale’ – пользователь указывает что изображение цветное, но надо преобразить в черно-белое.
8. +histogramGraph(type = ’grayscale’, format = ‘jpg’, quality = 90, coef = 35);  
   type – тоже самое что и у 12 пункта.  
   format – в каком формате будет изображение.  
   quality – в каком качестве будет изображение.  
   coef – коэффициент для размера графика, лучше оставлять по умолчанию.
9. +blur(value = 1);
10. +filters(filters) – принимает строку типа ‘opacity- contrast’ и применяет эти стили.
11. +sharpen(value = 10) – увеличивает четкость изображения  
    value – от 1 до 100
12. +text($text, $startX = 0, $startY = 0, $func) – вставляет текст на изображение.  
    func – callback функция для обработки цвета, размера, угл наклона и шрифта.
13. +outPut() – выводит изображение
14. +Save() – сохранить изображение в папку